UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Recuperação de imagens por conteúdo baseada em realimentação de relevância e classificador por floresta de caminhos ótimos

Autor: André Tavares da Silva

Orientador: Léo Pini Magalhães
Co-orientador: Alexandre Xavier Falcão

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Engenharia de Computação.
Aprovação em 26/07/2011

Banca Examinadora:
Prof. Dr. Clésio Luis Tozzi - UNICAMP
Prof. Dr. Léo Pini Magalhães - UNICAMP
Prof. Dr. Roberto de Alencar Lotufo - UNICAMP
Prof. Dr. Roberto Marcondes Cesar Junior - USP
Prof. Dr. Silvio Jamil Ferzoli Guimarães - PUC-Minas

Campinas, SP
2011

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA – BAE – UNICAMP

Silva, André Tavares da
Si38r Recuperação de imagens por conteúdo baseada em realimentação de relevância e classificador por floresta de caminhos ótimos / André Tavares da Silva. – Campinas, SP: [s.n.], 2011.

Orientadores: Léo Pini Magalhães; Alexandre Xavier Falcão.
Tese de doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.
1. Reconhecimento de padrões. 2. Recuperação de informação. 3. Análise de imagem. I. Magalhães, Léo Pini. II. Falcão, Alexandre Xavier. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês:	Content-based image retrieval based on relevance feedback and optimum-path forest classifier
Palavras-chave em Inglês:	Pattern recognition, Information retrieval, Image analysis
Área de concentração:	Engenharia de Computação
Titulação:	Doutor em Engenharia Elétrica
Banca examinadora:	Silvio Jamil Ferzoli Guimarães, Roberto Marcondes Cesar Junior, Roberto de Alencar Lotufo, Clésio Luis Tozzi
Data da defesa:	26/07/2011
Programa de Pós Graduação:	Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: André Tavares da Silva

Data da Defesa: 26 de julho de 2011

Título da Tese: "Recuperação de imagens por conteúdo baseada em realimentação de relevância e classificador por floresta de caminhos ótimos"

Prof. Dr. Léo Pini Magalhães (Presidente): _____

Prof. Dr. Silvio Jamil Ferzoli Guimarães: _____

Prof. Dr. Roberto Marcondes Cesar Junior: _____

Prof. Dr. Clésio Luis Tozzi: _____

Prof. Dr. Roberto de Alencar Lotufo: _____

Resumo

Com o crescente aumento de coleções de imagens resultantes da popularização da Internet e das câmeras digitais, métodos eficientes de busca tornam-se cada vez mais necessários. Neste contexto, esta tese propõe novos métodos de recuperação de imagens por conteúdo baseados em realimentação de relevância e no classificador por floresta de caminhos ótimos (OPF - *Optimum-Path Forest*), sendo também a primeira vez que o classificador OPF é utilizado em conjuntos de treinamento pequenos.

Esta tese denomina como guloso e planejado os dois paradigmas distintos de aprendizagem por realimentação de relevância considerando as imagens retornadas. O primeiro paradigma tenta retornar a cada iteração sempre as imagens mais relevantes para o usuário, enquanto o segundo utiliza no aprendizado as imagens consideradas mais informativas ou difíceis de classificar. São apresentados os algoritmos de realimentação de relevância baseados em OPF utilizando ambos os paradigmas com descritor único.

São utilizadas também duas técnicas de combinação de descritores juntamente com os métodos de realimentação de relevância baseados em OPF para melhorar a eficácia do processo de aprendizagem. A primeira, MSPS (*Multi-Scale Parameter Search*), é utilizada pela primeira vez em recuperação de imagens por conteúdo, enquanto a segunda é uma técnica consolidada baseada em programação genética.

Uma nova abordagem para realimentação de relevância utilizando o classificador OPF em dois níveis de interesse é também apresentada. Nesta abordagem é possível, em um nível de interesse, selecionar os pixels nas imagens, além de escolher as imagens mais relevantes a cada iteração no outro nível.

Esta tese mostra que o uso do classificador OPF para recuperação de imagens por conteúdo é muito eficiente e eficaz, necessitando de poucas iterações de aprendizado para apresentar os resultados desejados aos usuários. As simulações mostram que os métodos propostos superam os métodos de referência baseados em múltiplos pontos de consulta e em máquina de vetor de suporte (SVM). Além disso, os métodos propostos de busca de imagens baseados no classificador por floresta de caminhos ótimos mostraram ser em média 52 vezes mais rápidos do que os métodos baseados em máquina de vetor de suporte.

Palavras-chave: realimentação de relevância, recuperação de informação, análise de imagens, classificação de imagens, reconhecimento de padrões.

Abstract

Considering the increasing amount of image collections that result from popularization of the digital cameras and the Internet, efficient search methods are becoming increasingly necessary. In this context, this doctoral dissertation proposes new methods for content-based image retrieval based on relevance feedback and on the OPF (optimum-path forest) classifier, being also the first time that the OPF classifier is used in small training sets.

This doctoral dissertation names as “greedy” and “planned” the two distinct learning paradigms for relevance feedback taking into account the returned images. The first paradigm attempts to return the images most relevant to the user at each iteration, while the second returns the images considered the most informative or difficult to be classified. The dissertation presents relevance feedback algorithms based on the OPF classifier using both paradigms with single descriptor.

Two techniques for combining descriptors are also presented along with the relevance feedback methods based on OPF to improve the effectiveness of the learning process. The first one, MSPS (Multi-Scale Search Parameter), is used for the first time in content-based image retrieval and the second is a consolidated technique based on genetic programming.

A new approach of relevance feedback using the OPF classifier at two levels of interest is also shown. In this approach it is possible to select the pixels in images at a level of interest and to choose the most relevant images at each iteration at another level.

This dissertation shows that the use of the OPF classifier for content based image retrieval is very efficient and effective, requiring few learning iterations to produce the desired results to the users. Simulations show that the proposed methods outperform the reference methods based on multi-point query and support vector machine. Besides, the methods based on optimum-path forest have shown to be on the average 52 times faster than the SVM-based approaches.

Keywords: relevance feedback, information retrieval, image analysis, image classification, pattern recognition.

Dedicatória

Dedico esta tese aos meus **pais** e **irmãos**.

Dedico também à minha esposa **Claudia**, que é meu presente e futuro e que esteve todo o tempo ao meu lado.

Agradecimentos

Agradeço à minha família pela paciência e apoio incondicional durante o decorrer desta tese.

Ao meu orientador, Prof. Léo Pini Magalhães, pela atenção, paciência, compreensão, motivação e confiança que depositou em mim para que esse trabalho fosse desenvolvido.

Ao Prof. Alexandre Xavier Falcão, pelo apoio, confiança, criatividade e paciência que auxiliaram no desenvolvimento desta tese.

Aos colegas do Instituto de Computação que contribuíram com implementações e conjunto de dados utilizados no desenvolvimento deste trabalho.

Ao CNPq (processo 140968/2007-5) pelo apoio financeiro.

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

BoW - Bag of Words, página 18

$CBIR$ - Content-Based Image Retrieval / Recuperação imagens por conteúdo, página 2

CNS - Color Naming System, página 16

EDT - Euclidean Distance Transform / Transformada de Distância Euclidiana, página 15

$FFP4$ - Quarta função de adequação definida por Fan et al. (2004), página 36

FFT - Fast Fourier Transform / Transformada rápida de Fourier, página 14

$GOPF_{RF}$ - Relevance Feedback using OPF and Greedy learning / Realimentação de relevância usando OPF e aprendizado guloso, página 27

HSL - Hue, Saturation, Lightness / Matiz, Saturação, Luminosidade, página 16

IFT - Image Foresting Transform / Transformada Imagem-Floresta, página 24

$L1$ - Distância Manhatan ou Taxicab, página 9

$L2$ - Distância euclidiana, página 9

$MSPS$ - Multi-Scale Parameter Search, página 4

MST - Minimum Spanning Tree / Árvore Espalhada Mínima, página 23

OPF - Optimum-Path Forest / Floresta de Caminhos Ótimos, página 4

$OPF_{Bi-Level}$ - Relevance feedback using OPF and Bi-Level approach / Realimentação de relevância usando OPF e abordagem Bi-Level, página 27

OPF_{GP} - Relevance feedback using OPF and Genetic Programming / Realimentação de relevância usando OPF e Programação Genética, página 27

OPF_{MSPS} - Relevance feedback using OPF and MSPS / Realimentação de relevância usando OPF e MSPS, página 27

$P \times R$ - Curva Precisão vs. Revocação, página 47

$POPF_{RF}$ - Relevance Feedback using OPF and Planned learning / Realimentação de relevância usando OPF e aprendizado planejado, página 27

QEX - Query Expansion method, página 50

QPM - Query Point Movement, página 21

$Rel \times It$ - Curva percentual de imagens Relevantes retornadas por Iteração, página 48

RF - Relevance Feedback / Realimentação de relevância, página 3

SVM - Support Vector Machine / Máquina de Vetor de Suporte, página 22

SVM_{AL} - SVM Active Learning method, página 51

Lista de Símbolos

$\lambda(t)$ - Classe da imagem t (relevante ou irrelevante)
π_t - Um caminho (lista de nós) terminando em t
$\boldsymbol{\theta}$ - Conjunto de parâmetros da função de combinação do método MSPS
$\boldsymbol{\Delta}$ - Deslocamento em $\boldsymbol{\theta}$
$d(s,t)$ - Distância/diferença entre as imagens s e t
$\bar{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ - Distância média normalizada entre t e os protótipos relevantes e irrelevantes
$d_c(t,\mathcal{S}_R,\mathcal{S}_I)$ - Distância indicando a proximidade de t tanto às florestas relevantes quanto irrelevantes.
v - Função de extração de característica de uma imagem
$\vec{v}$ - Vetor de característica
C - Mapa de custo
$\mathcal{D}$ - Conjunto de descritores
δD - Função que combina os valores de distância
I - Número de iterações a executar
$\mathcal{L}$ - Conjunto de pixels de uma imagem
$\mathcal{M}$ - Conjunto de pixels marcados (objeto ou fundo)
$\mathcal{M}'$ - Conjunto de pixels marcados (objeto ou fundo) ordenados por custo
$\mathcal{N}_O$ - Lista de protótipos de objeto (pixels)
$\mathcal{N}_F$ - Lista de protótipos de fundo (pixels)
N - Número de imagens retornadas por iteração
P - Lista de predecessores da floresta de caminhos ótimos
R - Mapa de raízes
$\mathcal{R}$ - Lista de imagens retornadas pelo sistema como sendo relevantes
$\mathcal{S}_R$ - Lista de protótipos relevantes (em $\mathcal{T}$)
$\mathcal{S}_I$ - Lista de protótipos irrelevantes (em $\mathcal{T}$)
$\mathcal{T}$ - Conjunto de imagens marcadas pelo usuário como relevante/irrelevante durante as iterações
$\mathcal{T}'$ - Conjunto de treinamento ordenado por custo
$\mathcal{X}$ - Lista de imagens exibidas a cada iteração
$\mathcal{Y}$ - Lista de imagens classificadas como relevante pela OPF
$\mathcal{Z}$ - Base de imagens
$\mathcal{Z}\backslash\mathcal{T}$ - Imagens da base $\mathcal{Z}$ desconsiderando as imagens em $\mathcal{T}$ (complementar de $\mathcal{T}$ em relação a $\mathcal{Z}$)

Trabalhos Escritos Pelo Autor

1. A.T. Silva, A.X. Falcão, L.P. Magalhães. "A new CBIR approach based on relevance feedback and optimum-path forest classification". *Journal of WSCG*, 18 (1-3), pg. 73–80, 2010.

2. A.T. Silva, A.X. Falcão, L.P. Magalhães. "Active learning paradigms for CBIR systems based on optimum-path forest classification". *Pattern Recognition*, 44 (12) pg. 2971–2978, 2011.

3. J. A. dos Santos, A. T. da Silva, R. da S. Torres, A. X. Falcão, L. P. Magalhães, R. A. C. Lamparellic. "Interactive Classification of Remote Sensing Images by using Optimum-Path Forest and Genetic Programming". *Poster of the International Conference on Computer Analysis of Images and Patterns*.

4. A.T. Silva, J. A. dos Santos, A.X. Falcão, R. da S. Torres, L.P. Magalhães. "Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning". *Computer Vision and Image Understanding* (submetido em dezembro de 2010).

Capítulo 1

Introdução

Com o crescimento da internet e a popularização dos dispositivos para captura de imagens como câmeras digitais e *scanners*, a disponibilidade de coleções de imagens tem crescido rapidamente nos últimos anos. Por isso, os usuários necessitam cada vez mais de ferramentas eficientes para pesquisar, navegar e recuperar essas informações em diferentes domínios, como sensoriamento remoto, moda, prevenção de crime, publicidade, medicina, arquitetura, entre outros. Para este propósito, têm aumentado a importância de sistemas de recuperação de imagens.

A visão é sem dúvida um dos sentidos humanos mais desenvolvidos, assim como a capacidade do cérebro em interpretar as informações visuais. As pessoas são capazes de categorizar facilmente uma imagem ou parte dela, além de identificar objetos ou faces previamente conhecidos. Essa capacidade é excepcional e não existe ainda um sistema computacional capaz de realizar buscas de imagens com eficiência próxima a do sistema humano. Nos campos da psicologia e da neurociência, existem trabalhos que buscam simular computacionalmente áreas do cérebro envolvidas no processamento da informação visual (Kosslyn e Thompson, 2006; Shastri, 2001; Barkowsky et al., 2003; Palmeri e Gauthier, 2004; Ranganath, 2006; Kay et al., 2008), dentre os quais destaca-se o trabalho de Kosslyn (Kosslyn, 1980; Kosslyn e Thompson, 2006), que aborda esta questão de forma bastante detalhada e didática, propondo um modelo teórico para entender como o cérebro processa a informação visual. Outros trabalhos tentam reproduzir ou explicar fenômenos particulares (como ilusões visuais, entre outros) também através de sistemas computacionais (Glasgow e Papadias, 1992; Draper et al., 2004; Felsen e Dan, 2005; Davies et al., 2009). Esses trabalhos utilizam conceitos da psicologia e neurociência para simular computacionalmente áreas do cérebro (giro para-hipocampal, por exemplo) e testar a validade de modelos teóricos sobre alguma funcionalidade cerebral. Outros visam contribuir na área de visão computacional (Croft e Thagard, 2002; Gorder, 2008; Serre e Poggio, 2010) como o trabalho seminal de Marr (1982) em visão computacional que inspirou muitos conceitos da área de neurociência e de neuroanatomia. Assim não existe um consenso (Churchland et al., 1994;

Pylyshyn, 2000; Thomas, 2010) ou um entendimento completo a respeito de como o cérebro processa as imagens, e por isso mesmo o desenvolvimento de um sistema computacional de busca de imagens que tenha uma performance similar ao processo visual humano continua sendo um desafio.

Nos sistemas computacionais existentes para recuperação de imagens oriundos de áreas de pesquisa relacionadas à informática, existem dois paradigmas principais: baseado em texto e em conteúdo. As abordagens baseadas em texto começaram a ser utilizadas na década de 1970. Nesses sistemas, o processo de recuperação consiste em comparar os termos de uma consulta textual, definida por um usuário, com as anotações associadas às imagens por palavras-chave e, a partir dessa comparação, retornar um conjunto de imagens. Existem duas desvantagens principais nesta abordagem: a necessidade de um trabalho humano considerável para realizar as anotações e a imprecisão das anotações devido a ambiguidade e imprecisão das palavras utilizadas na anotação pelas pessoas. O conteúdo de uma imagem normalmente é muito mais rico do que é possível descrever através de um conjunto de palavras (Datta et al., 2008; Wang et al., 2010).

Os sistemas de recuperação de imagens baseados em conteúdo (CBIR - *Content-Based Image Retrieval*) (Smeulders et al., 2000; Liu et al., 2007; Datta et al., 2008; Vasconcelos, 2001; Snoek e Smeulders, 2010) têm sido estudados e propostos para tentar superar essas desvantagens de sistemas de recuperação de imagens baseados em texto. Nos sistemas CBIR, as imagens são indexadas pelo seu conteúdo visual, tal como cor, textura e forma, tornando desnecessária a anotação manual. Os métodos de extração de característica de imagens utilizam descritores, que representam o conjunto de pixels de uma imagem através de um vetor de característica, que expressa componentes da imagem relacionadas por exemplo a cor, ou forma, ou textura, etc. O processo de busca consiste basicamente em, dado um padrão de consulta (normalmente uma imagem), calcular a sua similaridade em relação às imagens armazenadas em uma base de imagens e exibir as mais similares. Esta similaridade é obtida comparando os descritores da imagem de consulta com os descritores associados às imagens da base de interesse. De resultados da área de neurociência, sabe-se que a chave para o reconhecimento de objetos é o sistema ser capaz de discriminar objetos sendo tolerante a transformações de rotação, escala, translação, iluminação, mudança de ponto de vista e organização. Por isso, os descritores que buscam traduzir as propriedades visuais utilizadas para descrição das imagens devem ser invariantes a essas transformações.

Existem muitos esforços para encontrar o melhor descritor para um dado problema ou determinar como diferentes características podem ser combinadas. Diversas técnicas de aprendizado são utilizadas para este fim. A grande maioria dos trabalhos nesta área aborda problemas específicos, como imagens médicas, sensoriamento remoto, impressão digital, entre outros. Normalmente os sistemas são utilizados pelos próprios desenvolvedores ou especialistas. Isto porque é possível determinar um descritor ou um conjunto de descritores específicos para um determinado fim, tornando a consulta de

imagens em um problema de reconhecimento de padrões.

No entanto, a percepção visual dos objetos é subjetiva e por isso não existe uma única característica capaz de representar todas as consultas de imagens para propósito geral. O uso da realimentação do usuário a partir da relevância da resposta obtida, denominada de *realimentação de relevância* (RF - *Relevance Feedback*) tem sido bastante utilizada juntamente com técnicas de aprendizado para recuperação de imagens por conteúdo. O usuário informa quais as imagens ele considera relevantes em um conjunto de imagens retornado pelo sistema e o algoritmo de realimentação de relevância aprende a vontade do usuário durante um determinado número de iterações. Dessa forma, o sistema retorna imagens cada vez mais similares à vontade do usuário, aprendendo o conceito estabelecido por ele. Uma vantagem desta técnica é que ela possibilita ao usuário expressar sua necessidade na especificação da consulta sem que ele precise conhecer a fundo as propriedades de representação da imagem.

Ao longo das últimas décadas, diversos produtos comerciais e protótipos experimentais para recuperação de imagens por conteúdo foram desenvolvidos, como IBM QBIC (Faloutsos et al., 1994), MIT Photobook (Pentland et al., 1996), Virage (Gupta e Jain, 1997), Columbia VisualSEEK e WebSEEK (Smith e Chang, 1996), UCSB NeTra (Ma e Manjunath, 1997), Berkeley Chabot (Ogle e Stonebraker, 1995), UIUC MARS (Mehrotra et al., 1997), Stanford WBIIS (Wang et al., 1998), PicHunter (Cox et al., 2000), SIMPLIcity (Wang et al., 2001), PicToSeek (Gevers e Smeulders, 2000), Blobworld (Carson et al., 2002), CIRES (Iqbal e Aggarwal, 2002), LTU-Corbis (Veltz, 2004), Fids (Li et al., 2005), CORTINA (Gelasca et al., 2007), TinEye (Boujnane e Bloore, 2009) e Windsurf (Bartolini et al., 2010). Este ainda é um problema em aberto e de difícil solução, como é mostrado através dos trabalhos da Seção 2.2.

Existem basicamente três categorias de sistemas em relação ao objetivo do usuário na busca de imagens por conteúdo. Na primeira categoria, o usuário visa buscar *uma imagem específica*. Neste caso, o usuário sabe exatamente como e qual é a imagem que ele está buscando e a pesquisa termina quando esta imagem é encontrada. Na categoria de *busca por abrangência*, o sistema busca todas as imagens que estão a uma certa distância da imagem de consulta. A categoria mais comum é a *busca por similaridade*, na qual é especificado o número de imagens que devem ser retornadas que são mais similares ao padrão de consulta. Neste trabalho, apesar de ser possível endereçar qualquer uma das categorias apresentadas acima, os experimentos foram realizados em busca por similaridade para encontrar um conjunto de imagens semelhantes ao padrão apresentado como imagem de consulta inicial.

Desta forma, o principal objetivo desta tese é discutir modelos capazes de permitir que um usuário comum consiga, com um mínimo de iterações, encontrar imagens de seu interesse. Segundo Serre e Poggio (2010), o reconhecimento de imagens do ponto de vista da neurociência é sinônimo de iden-

tificação ou categorização e ambos, do ponto de vista computacional, envolvem *classificação*. Por isso acreditamos que modelos baseados em classificadores são o caminho mais adequado para obter a melhor eficiência na recuperação de imagens. Assim, após um estudo do estado da arte na área, esta tese propõe novos modelos e técnicas baseadas principalmente na classificação e ordenação das imagens de uma base objetivando a recuperação de imagens para propósitos gerais. Estas técnicas utilizam realimentação de relevância e um método de classificação de padrões muito rápido e eficiente baseado em floresta de caminhos ótimos (OPF – *Optimum-Path Forest*) (Papa et al., 2009). Com o conjunto de imagens rotuladas, a cada iteração, pelo usuário como relevante ou não, o método gera uma floresta de caminhos ótimos e somente as imagens classificadas como sendo relevantes são ordenadas por distância e apresentadas ao usuário na próxima iteração. Essa ordenação é calculada através de uma distância média normalizada para os protótipos da OPF. Através de diversos experimentos, é demonstrado que esta estratégia é realmente muito eficaz, reduzindo consideravelmente o número de iterações necessárias para retornar as imagens requeridas pelo usuário, permitindo assim implementar um sistema CBIR eficaz e de alta eficiência.

Esta tese sugere a existência de dois paradigmas de aprendizado distintos na busca por similaridade em relação às imagens retornadas a cada iteração, aqui denominados de guloso e planejado. No paradigma guloso, sempre as imagens mais prováveis de serem relevantes, ou seja, as N imagens mais próximas do padrão de consulta são apresentadas ao usuário. No planejado, o aprendizado é realizado através das N imagens consideradas mais informativas ou difíceis de se classificar. São apresentados os algoritmos utilizando ambos os paradigmas de realimentação de relevância usando o classificador OPF.

Conforme mencionado anteriormente, o conteúdo das imagens pode ser representado por múltiplas características. Encontrar a melhor função de combinação de descritores é um problema de otimização. Desta forma, foram desenvolvidas duas abordagens para combinação de descritores, MSPS (*Multi-Scale Parameter Search*) (Ruppert et al., 2010) e programação genética, permitindo mostrar que o uso do classificador OPF pode ser ainda mais efetivo.

O conteúdo de uma imagem (região ou objeto) pode ser melhor especificado com a marcação de pixels relevantes e irrelevantes. Desta forma, o classificador OPF pode também ser utilizado para selecionar o conteúdo relevante em uma imagem. Esta tese apresenta também uma nova abordagem para realimentação de relevância utilizando o classificador OPF em dois níveis de interesse, tanto para escolher as imagens mais relevantes a cada iteração quanto para selecionar os pixels de interesse nas imagens.

Esta tese está estruturada da seguinte maneira. O Capítulo 2 apresenta conceitos sobre a área de CBIR e trabalhos relacionados à área de realimentação de relevância, abordando os fundamentos mais importantes para o desenvolvimento deste trabalho. Os métodos propostos nesta tese são apresentados

no Capítulo 3, enquanto os resultados obtidos são mostrados no Capítulo 4. Por fim, no Capítulo 5 são comentadas as contribuições para o estado da arte assim como as perspectivas futuras para o presente trabalho.

Capítulo 2

Aspectos relacionados à recuperação de imagens

Como já exposto, a quantidade de coleções de imagens digitais têm crescido continuamente nos últimos anos, na Internet por exemplo nas redes sociais, Flickr[1] e Picasa[2] e a maneira mais usual para recuperação de imagens pela Internet ainda utiliza as informações textuais obtidas da Web (Cai et al., 2004; Feng et al., 2004) conjuntamente com as imagens. Embora esta seja a abordagem mais empregada pelos usuários, pesquisas de recuperação de imagens por conteúdo têm crescido consideravelmente nos últimos anos, como pode ser constatado ao longo desta tese.

A percepção visual dos objetos pelas pessoas é subjetiva e por isso é difícil representar uma imagem através de uma única característica. A *cor* é provavelmente a característica mais utilizada para recuperação visual, sendo relativamente robusta por apresentar independência do tamanho e da orientação da imagem. Além disso, muitos trabalhos se utilizam desta característica juntamente com conceitos de alto nível, por exemplo, "céu claro" pode ser definido como uma imagem com a região superior uniformemente azul. *Textura* é outra propriedade presente em praticamente todas as estruturas, como nuvens, vegetação, paredes, cabelo e outras, contendo informação importante sobre o arranjo estrutural da superfície. A *forma* é também uma importante informação utilizada pelas pessoas para reconhecimento de objetos. A forma é utilizada em aplicações como reconhecimento de caracteres, detecção e reconhecimento de pessoas em sistemas de segurança e rastreamento de objetos em vídeo. Assim, para a recuperação de imagens, cor, textura e forma devem ser extraídas de uma imagem e expressas através de um *vetor de característica*. Esse vetor de característica pode ser interpretado como um ponto no espaço $\mathcal{R}^n$, onde n é o tamanho do vetor de característica e a busca das imagens será feita, por exemplo, baseada na comparação da distância entre o vetor de

[1] http://www.flickr.com
[2] http://picasa.google.com

característica da imagem de consulta e os vetores de característica de imagens de uma base.

Será adotada nesta tese a definição de *descritor* de imagem como a junção do resultado de um algoritmo de extração de característica e da função de distância entre vetores. A extração de característica codifica o conteúdo de uma imagem no vetor de característica, enquanto a função de distância define a similaridade entre dois vetores de característica e, consequentemente, entre duas imagens. Essa similaridade é dada pelo inverso da função de distância, ou seja, quanto menor a distância entre os vetores de característica das imagens, maior a similaridade entre elas. Os descritores também podem ser definidos somente pela função de extração de característica, mas a definição adotada nesta tese permite que diferentes características de cor, textura e forma sejam codificadas em vetores diferentes e com métricas de comparação diferentes.

Formalmente, sendo $\mathcal{Z}$ um banco de imagens, cada imagem $t \in \mathcal{Z}$ é representada por um descritor composto por um vetor de característica $\vec{v}(t)$, calculado através da função de extração de característica v e pela função de distância $d(s,t)$ entre os vetores de característica das duas imagens $s,t \in \mathcal{Z}$. Um descritor, então é uma tupla (v,d) formada pela função de extração de característica e pela função de distância. Uma imagem pode também ser representada por mais de um descritor como pode ser visto na Seção 3.3.

Quando um descritor captura a informação contida na imagem inteira, ele é chamado de *descritor global*. Caso as característica das imagens sejam extraídas para diferentes partes (regiões, bordas ou pontos de interesse) é denominado de *descritor local*. Os vetores de característica dos descritores locais são calculados sobre as partes, como por exemplo as regiões em torno de pontos de interesse. Desta forma, se uma imagem sofre uma deformação geométrica ou a cena é observada de outro ângulo de visão, a correspondência entre as mesmas características (ou pontos) pode ser encontrada. Os descritores locais mais utilizados atualmente são SIFT (Lowe, 1999) e SURF (Bay et al., 2008). Apesar de ser possível utilizar tanto descritores locais quanto globais nesta tese, são apresentados apenas resultados obtidos usando descritores globais.

2.1 Descritores de imagem

A extração de características é a base da recuperação de informação visual por conteúdo, tendo como objetivo obter os atributos de mais baixo nível de uma imagem. Estas características podem ser classificadas como sendo de domínio geral ou de domínio específico. O primeiro domínio inclui características gerais de cor, textura e forma, enquanto o último é dependente da aplicação, como por exemplo, classificação de impressão digital, placas de veículos ou faces. As características de domínio específico são descritas na literatura que trata do padrão a ser reconhecido e envolvem o conhecimento de características particulares do problema em questão (detecção de minúcias na impressão digital,

por exemplo). Devido ao fato dos problemas testados (Seção 4.1) necessitarem de métodos para busca de imagens de domínio geral, foram utilizados descritores com características gerais.

Excelentes pesquisas sobre descritores podem ser encontradas nos trabalhos de Tuytelaars e Mikolajczyk (2008), Rui et al. (1999) e de Penatti (2009). Esta seção apresenta alguns descritores de imagens utilizados nos resultados do Capítulo 4. É descrito, resumidamente, como é obtido um vetor de característica e quais as métricas usadas para cálculo de similaridade entre duas imagens. A função de distância utilizada em diversos trabalhos é a distância euclidiana, também conhecida por distância L2 (Equação 2.2).Por outro lado, alguns trabalhos (Penatti, 2009) mostram que diversos descritores conseguem obter melhores resultados utilizando a função de distância L1 (também conhecida por distância Manhatan ou Taxicab), onde a distância entre dois pontos é dada pela soma das diferenças absolutas entre as suas coordenadas (Equação 2.1). Alguns descritores definem uma métrica própria para calcular a distância entre dois vetores de característica (como no descritor *Color Bitmap* apresentado na Seção 2.1.1). Nesta tese foram utilizadas implementações de descritores de imagens desenvolvidas no LIV (Laboratório de Informática Visual)[3] do Instituto de Computação da Unicamp. A seguir são apresentados os descritores baseados em cor, textura e forma utilizados para geração dos resultados dos experimentos do Capítulo 4.

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i| \tag{2.1}$$

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.2}$$

2.1.1 Descritores baseados em cor

- ACC (Huang et al., 1997)

  Huang et al. (1997) definem um descritor de imagens chamado de correlograma de cor (*color correlogram*), que calcula a correlação espacial entre as cores dos pixels de uma imagem. Um correlograma de cor indica a probabilidade de se encontrar na imagem dois pixels de mesma cor a uma certa distância um do outro. O auto-correlograma (*autocorrelogram*) é um subconjunto do correlograma que captura a correlação espacial somente entre cores idênticas, e o descritor de imagem baseado no auto-correlograma de cor é chamado de ACC (do inglês *Auto Color Correlation*, conforme Penatti (2009)) nesta tese. Seu algoritmo de extração realiza inicial-

[3] http://www.liv.ic.unicamp.br

mente uma quantização do espaço de cor. Dada uma imagem I, o conjunto I_c armazena os pixels p de uma mesma cor c da imagem I. Para cada distância k, percorre-se a imagem a fim de calcular o auto-correlograma (Equação 2.3).

$$\alpha_c^k = \quad Pr[p_2 \in I_c || p_1 - p_2| = k], p_1 \in I_c, p_2 \in I. \tag{2.3}$$

Ao final têm-se m valores de probabilidade para cada distância k, e portanto, o tamanho do vetor de característica é $m \times k$. Nos resultados apresentados no Capítulo 4 foram utilizados os mesmos parâmetros do artigo original (Huang et al., 1997): quatro valores de distância (1, 3, 5 e 7) e o espaço de cor RGB quantizado em 64 *bins* (faixas de valores). A função de distância entre as imagens é definida no trabalho original por uma variação da função L1, onde a subtração de cada termo é normalizada pela soma dos valores correspondentes. Na implementação utilizada (Penatti, 2009), também foi adotada a função de distância L1.

- BIC (*border/interior pixel classification*) (Stehling et al., 2002)

  Descritor de cor que classifica os pixels da imagem em pixels de borda ou pixels de interior. Um pixel que possui a mesma cor dos seus vizinhos (vizinhança 4) é classificado como interior e os outros são classificados como borda. São calculados dois histogramas de cor, sendo um para os pixels de borda e outro para os pixels de interior. Os dois histogramas são concatenados e armazenados como um único histograma. Nesta tese o descritor BIC utiliza 64 faixas de valores de cor RGB para borda e interior, resultando em um vetor de característica de 128 elementos. A comparação dos histogramas é feita utilizando-se a distância $dLog$, extraindo o logaritmo na base 2 para cada posição do vetor de característica antes de se calcular a distância usando L1.

- *Color Bitmap* (Lu e Chang, 2007)

  O descritor de imagem por cor baseado no método *image bitmap feature* proposto por Lu e Chang (2007) calcula o vetor de característica através das cores dos pixels no espaço RGB na imagem inteira (globalmente) e por regiões fixas, sendo chamado nesta tese *Color Bitmap*. Média e desvio padrão são calculados para cada espaço R, G e B separadamente para todos os pixels da imagem. A imagem é então dividida em blocos sem sobreposição e a média dos valores de cada bloco é calculada. As médias dos blocos são comparadas à média da imagem completa e, se a média do bloco for menor que a média da imagem, o elemento do vetor de característica para o bloco recebe valor 1. Caso contrário, recebe valor 0. Portanto, para RGB, o vetor de característica é formado por três valores binários para cada um dos blocos. Além disso, também são armazenados as médias e desvio padrão para os três canais de cor, sendo então mais seis valores. As imagens nesta tese foram divididas em 100 blocos (10×10) e,

por isso, o vetor de característica possui 300 bits mais seis valores. A função de distância é calculada em duas etapas. Na primeira, é calculada a distância euclidiana (L2) entre os valores globais (média e desvio padrão). Na segunda, usa-se a distância de Hamming entre os valores binários de cada bloco. Deve-se lembrar que a distância de Hamming de uma sequência binária a outra de mesmo comprimento é a quantidade de termos que diferem entre si, ou seja, é igual ao número de uns ao calcular XOR entre as duas sequências.

- JAC (Williams e Yoon, 2007)

  O descritor de cor aqui denominado JAC (*Joint auto-correlogram*) segue o princípio do descritor ACC, mas calcula o correlograma conjunto para mais de uma propriedade da imagem: cor, magnitude do gradiente, *rank* e *texturedness*. A cor é calculada pela quantização do espaço RGB, enquanto as outras propriedades são calculadas a partir da imagem em escala de cinza. A magnitude do gradiente é calculada como o máximo entre a diferença de brilho do pixel atual e seu vizinho da direita e de seu vizinho de baixo. *Rank* é representado pela quantidade de pixels da vizinhança que tem brilho maior do que o brilho do pixel atual. *Texturedness* é calculada contando-se a quantidade de pixels na vizinhança com diferença de brilho maior do que um certo limiar em relação ao pixel atual. Um auto-correlograma conjunto indica, para cada distância, a probabilidade de ocorrerem simultaneamente as quatro propriedades consideradas. O espaço RGB foi quantizado nesta tese por 64 valores de cor, enquanto cada uma das propriedades de magnitude do gradiente, *rank* e *texturedness* foram quantizados em 5 valores. Foram usados quatro valores de distância para o cálculo do auto correlograma: 1, 3, 5 e 7. O vetor de característica contém todas as possíveis combinações entre os valores das quatro propriedades para cada um dos valores de distância usados. Assim, o tamanho do vetor de característica é de 32.000 elementos (64*5*5*5*4). Neste descritor também foi utilizada a função de distância L1.

2.1.2 Descritores baseados em textura

- LAS (local activity spectrum) (Tao e Dickinson, 2000)

  Captura a atividade espacial de uma textura nas direções horizontal, vertical, diagonal e anti-diagonal separadamente. O algoritmo de extração de característica extrai as informações de textura usando 4 medidas de atividade espacial para cada pixel (i, j) de uma imagem, calculados por:

$$g_1 = \mid f(i-1, j-1) - f(i,j) \mid \; + \mid f(i,j) - f(i+1, j+1) \mid, \tag{2.4}$$

$$g_2 = \mid f(i-1, j) - f(i,j) \mid \; + \mid f(i,j) - f(i+1, j) \mid, \tag{2.5}$$

$$g_3 = \mid f(i-1,j+1) - f(i,j) \mid \; + \mid f(i,j) - f(i+1,j-1) \mid, \tag{2.6}$$

$$g_4 = \mid f(i,j+1) - f(i,j) \mid \; + \mid f(i,j) - f(i,j-1) \mid. \tag{2.7}$$

Onde $f(i,j)$ é o valor da cor para pixel em (i,j). Estas medidas são usadas para o cálculo de um histograma chamado *local activity spectrum*, cujos componentes foram quantizados em 4 valores na implementação utilizada nesta tese, gerando um histograma de 256 valores. A função de distância do descritor LAS é a função L1.

- LBP (local binary pattern) (Takala et al., 2005)

  É um descritor de textura que extrai informações de variação dos brilhos entre pixels vizinhos. Para isso, é definida uma janela com raio r e uma quantidade de vizinhos p. A imagem é percorrida considerando as variações entre os brilhos dos pixels vizinhos em relação ao pixel central da janela, sendo 1 para variação positiva e 0 para negativa. Para garantir invariância à rotação, é feita a contagem das transições entre 0/1 e 1/0. Se a quantidade for menor ou igual a 2, o valor LBP é a quantidade de sinais 1. Senão, o valor LBP é $p+1$. O vetor de característica é um histograma contendo $p+2$ valores. A implementação feita por Pennati (Penatti, 2009), usada nesta tese, utiliza raio unitário ($r = 1$) de vizinhança 8 ($p = 8$), calculando a distância entre seus vetores de característica através da função L1.

- SASI (*Statistical Analysis of Structural Information*) (ÇarkacIoglu e Yarman-Vural, 2003)

  Codifica as propriedades de uma imagem baseado em suas propriedades estruturais da textura usando janelas com diferentes resoluções e orientações. O primeiro passo do algoritmo é escolher os tamanhos das janelas e as orientações a serem usadas. Cada janela pode ser percorrida de diferentes maneiras, que são determinadas por vetores chamados de *lag vectors*. A quantidade de *lag vectors* por janela depende da largura K da janela. K é obtido pela equação $K = \lceil (S/4) \rceil + 1$, onde S é a largura da janela em pixels. Para cada janela, o algoritmo percorre a imagem calculando um valor de auto-correlação considerando diferentes direções. Ao final, tem-se uma imagem de valores de auto-correlação para cada configuração de cada janela e o vetor de característica armazena os valores de média e desvio padrão de cada janela. Os valores de média e desvio padrão do vetor de característica são então normalizados pela média e desvio padrão globais da base de imagens usada. Como não é comum que se conheça a priori todas as imagens a serem procuradas, na implementação utilizada esta normalização é feita pela média e desvio padrão dos valores do vetor de característica. Foram utilizadas nos experimentos janelas de tamanhos 3x3, 5x5 e 7x7 pixels em 4 direções (0^o, 45^o, 90^o e 135^o). O tamanho do vetor de característica gerado é de 64 elementos ($2 \cdot 4 \cdot (\lceil \frac{3}{4} \rceil + 1 + \lceil \frac{5}{4} \rceil + 1 + \lceil \frac{7}{4} \rceil + 1)$). A função de distância usada pelo SASI calcula um valor de similaridade por meio de produtos internos

entre dois vetores de característica.

- SID (*Invariant Steerable Pyramid Decomposition*) (Montoya-Zegarra et al., 2007)

  No descritor aqui denominado SID, a imagem é processada por um conjunto de filtros sensíveis à escala e orientação. Inicialmente a imagem é decomposta por uma *wavelet* em duas sub-bandas usando um filtro passa-altas e um filtro passa-baixas. A banda resultante do filtro passa-baixas é decomposta recursivamente em k sub-bandas por filtros passa-bandas e em uma sub-banda por um filtro passa-baixas. Para cada escala e, são calculadas k informações direcionais nas etapas da recursão, resultando em $e * k$ sub-bandas resultantes. A média e o desvio padrão de seus valores são armazenados no vetor de característica. Para obter invariância à rotação é encontrada a orientação dominante somando as energias[4] das escalas para cada orientação. Quanto maior a soma das energias, mais dominante é a orientação. O vetor de característica é deslocado circularmente a fim de deixar a orientação dominante no início do vetor para cada escala. O mesmo pode ser feito para obter invariância à escala. Na implementação utilizada nesta tese, foram utilizadas 3 escalas e 6 orientações criando um vetor de característica invariante à escala e à orientação. A função de distância soma as diferenças entre as médias e desvios padrão correspondentes, aplicando uma normalização por valores relativos à base de imagens usada. O trabalho de Iniciação Científica da aluna Thalita Drumond (Drumond e Magalhães, 2010) mostrou que para as bases utilizadas nesta tese, essa normalização não resultou em ganho de eficácia. Por isso, foi utilizada a função de distância L1 sem a normalização para comparar dois vetores de característica.

2.1.3 Descritores baseados em forma

Para o uso de descritores de forma, as imagens precisam estar segmentadas. Como o processo de segmentação é complexo e existem diferentes técnicas para fazer isso, normalmente as propostas de uso de forma não contemplam esta fase e utilizam imagens já segmentadas. Os descritores aqui relacionados utilizam imagens binárias (em preto e branco) para a geração dos vetores de característica. Nos resultados do Capítulo 4 foram utilizadas bases de imagens previamente segmentadas.

- Fourier (Zhang e Lu, 2003)

  Neste descritor, os coeficientes da transformada de Fourier do contorno da imagem geram os valores dos vetores de característica para descrever um objeto, representando a forma no domínio da frequência. Os coeficientes de menor frequência contêm informações sobre as características gerais da forma, enquanto os de maior frequência contêm informações sobre os detalhes.

[4]Densidade espectral, PSD (*power spectral density*), ou ESD (*energy spectral density*)

Embora o número de coeficientes gerados a partir da transformação seja geralmente grande, um pequeno subconjunto dos coeficientes é suficiente para captar as características gerais da forma, pois as informações de frequência mais altas podem ser ignoradas, já que descrevem detalhes que não são tão úteis na discriminação da forma. Normalmente, os contornos que representam diferentes objetos possuem tamanhos diferentes e consequentemente diferentes quantidades de pixels representarão os objetos. Para garantir que os vetores de característica tenham um mesmo tamanho para fim de comparação, a assinatura de forma de todos os objetos deve ter o mesmo tamanho. Para facilitar o uso da FFT, recomenda-se utilizar tamanhos em potência de dois. Quanto maior o número de pontos, maior a precisão do descritor mas menor a eficiência computacional. Uma assinatura de forma é uma função unidimensional que representa áreas ou contornos. No descritor utilizado nesta tese, foram definidas quatro assinaturas de forma: distância ao centróide, coordenadas complexas (em função da posição), curvatura e função angular cumulativa. Para uma dada assinatura de forma $s(t)$ de tamanho N ($t = 0, 1, \ldots, N-1$), a transformada rápida de Fourier é calculada pela Equação 2.8.

$$u_n = \frac{1}{N} \sum_{t=0}^{N-1} s(t) exp(\frac{-j2\pi nt}{N}), n = 1, 2, \ldots, N-1 \quad (2.8)$$

Os coeficientes u_n, $n = 0, 1, \ldots, N-1$, são os valores do vetor de característica da forma. É usada a distância L2 (euclidiana) para comparar dois vetores de característica neste descritor.

- MSF (MultiScale Fractal) (Torres et al., 2004)

  Se a forma for usada como descrição de um objeto em uma cena cuja distância de visualização é variável, uma estrutura multi-escala é vantajosa para relacionar os vários pontos de vista, tornando a representação do objeto invariante em relação à distância de visualização. Enquanto a dimensão topológica em uma imagem é restrita a valores inteiros, a dimensão fractal permite o uso de valores fracionários. A dimensão fractal multi-escala de uma forma é calculada baseada na transformada de distância euclidiana (EDT - *Euclidean Distance Transform*) (Ragnemalm, 1993) dos pixels, que é relacionada ao diagrama de Voronoi. Cada pixel do objeto define uma área de influência (regiões de Voronoi discretas) composta pelos pixels mais próximos na imagem. Dado um conjunto S de pontos, representados em termos de suas coordenadas cartesianas (x, y), a sua dilatação euclidiana por um raio r, representada como S_r, é definida como sendo a união de todos os discos de raio r centrados em cada um dos pontos de S. Observa-se que, esta definição é válida tanto para objetos discretos quanto contínuos. Dilatações subsequentes de uma forma dada por valores crescentes de r criam conjuntos de formas cada vez mais simplificadas do objeto original. Sendo a forma representada em termos do conjunto S, $A(r)$ é a

área da versão dilatada da forma, e portanto, a dimensão fractal F é definida pela Equação 2.9.

$$F = 2 - \lim_{r \to 0} \frac{\log(A(r))}{\log(r)} \tag{2.9}$$

Considerando a imagem em um espaço bidimensional, sua dimensão fractal é um número entre 0 e 2. O vetor de característica usado nesta tese é composto por 50 elementos e são comparados usando a distância euclidiana (L2).

- TSDIZ (*Tensor Scale Descriptor with Influence Zones*) (Andaló et al., 2010)

  Escala tensorial (*Tensor Scale*) (Miranda et al., 2005), em qualquer ponto da imagem, é a representação paramétrica da maior elipse centrada neste ponto contida em uma mesma região homogênea, de acordo com um critério pré-determinado (no caso de imagens binárias, os pixels de mesma cor). A elipse é definida por três fatores: orientação, anisotropia e espessura. A orientação é representada pelo ângulo entre o eixo horizontal na imagem e o maior eixo da elipse, a anisotropia é calculada a partir dos comprimentos dos semi-eixos maior e menor e a espessura é representada pelo semi-eixo menor. Para extrair o vetor de característica de uma imagem, primeiramente é obtido o *Tensor Scale* para todos os pixels do objeto (região interior da imagem binária) obtendo-se, assim, os valores de anisotropia, orientação e espessura das elipses centradas em cada pixel da região interior. A seguir, o contorno é dividido em um número n_s predefinido de segmentos de mesmo tamanho. Cada segmento recebe um rótulo diferenciado, seguindo a ordem em que estão dispostos no contorno. Em um terceiro passo, o TSDIZ mapeia a média angular das orientações das elipses encontradas nas zonas de influência correspondentes a cada segmento no contorno. Para determinar a qual segmento pertence cada elipse é usado um mapa de rótulos calculado pela transformada de distância euclidiana (EDT). O vetor de característica é formado por n_s médias angulares, na ordem em que são mapeadas para os segmentos do contorno. Em vez de utilizar uma comparação exaustiva do vetor de característica, a implementação utilizada faz uso de uma função mais simples calculada pela média das orientações das elipses *Tensor Scale* contidas no vetor de característica.

2.2 Recuperação de imagens

O diagrama apresentado na Figura 2.1 resume as diversas técnicas para recuperação de imagens e as caixas em cinza destacam aquelas desenvolvidas nesta tese. Estas técnicas buscam reduzir a lacuna semântica entre as características de baixo nível (descritores) e o desejo do usuário, através do uso de técnicas de aprendizado. Muitos sistemas combinam mais de uma técnica para recuperação de

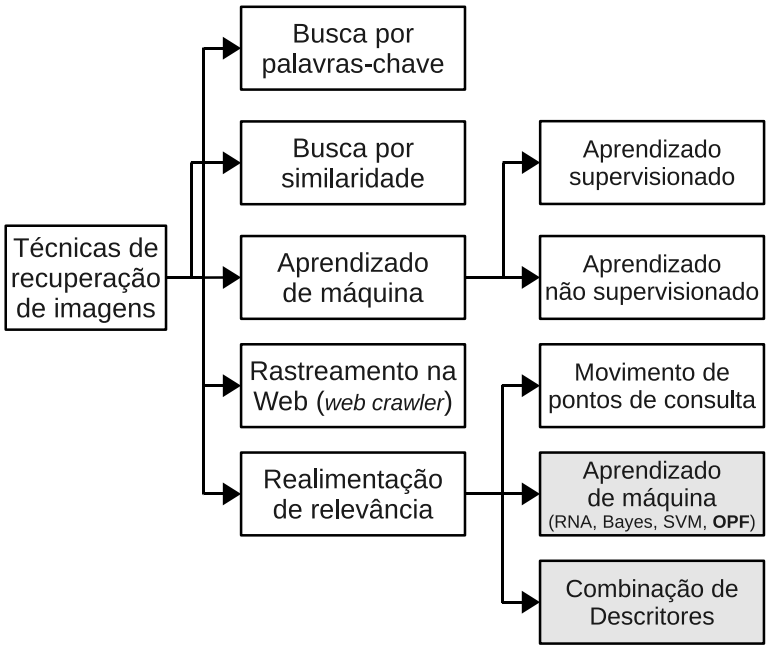


Fig. 2.1: Principais técnicas para recuperação de imagens.

imagem, como por exemplo sistemas de realimentação de relevância que usam busca por similaridade em um primeiro momento. Alguns sistemas de realimentação de relevância também utilizam aprendizado não supervisionado para agrupamento de imagens em uma etapa de pré-processamento a fim de aumentar a eficiência da busca em bases muito grandes.

Como comentado anteriormente, o uso de anotações de palavras-chave foi a primeira abordagem para a recuperação de imagens. Cada imagem tem a si associadas uma ou mais palavras-chave para identificar seu conteúdo e que foram anotadas previamente. A busca de imagens é feita retornando aquelas que possuem as expressões desejadas pelo usuário. Para evitar o esforço de anotações em bases de imagens grandes, foram definidas formas automáticas para associar nomes a conceitos de baixo nível (Mezaris et al., 2003; Liu et al., 2004). A atribuição de valores ou nomes para propriedades de cor e textura são fundamentais nesse tipo de sistema. Um sistema de nomenclatura de cores utilizado para este fim é o CNS (*Color Naming System*) proposto por Berk, Brownston, e Kaufman (1982), distinguindo 627 cores diferentes definidas no espaço HSL (*Hue*, *Saturation* e *Lightness* - Matiz, Saturação e Luminosidade). Matiz e luminosidade são quantizados em diferentes faixas de valores, onde a matiz é usada para definir o nome da cor e a luminosidade para a intensidades (por exemplo, *black*, *gray* ou *white*).

Devido ao excessivo esforço para a anotação em bases de imagens muito grandes, surgiram no início da década de 90 os primeiros sistemas para busca de imagens através de conteúdo. Recuperar

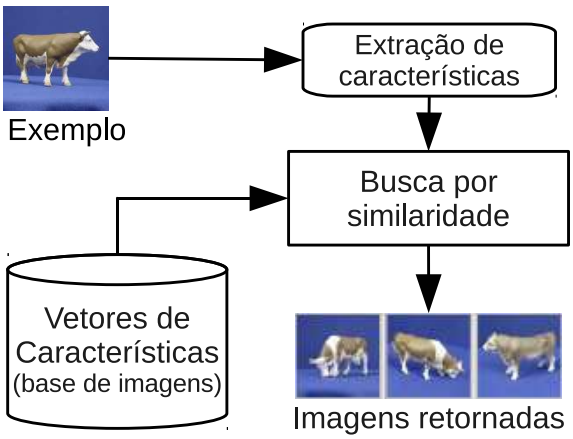


Fig. 2.2: Busca por similaridade.

imagens de uma base desta forma requer a construção de descritores de imagem e de técnicas para utilizá-los na recuperação de imagens.

Na busca por similaridade (*query by example*), uma imagem inicial é utilizada como entrada para o sistema, servindo de base para a pesquisa. O algoritmo de pesquisa varia conforme a aplicação, mas o objetivo é que as imagens resultantes devem compartilhar elementos ou características em comum com a imagem original. Neste tipo de pesquisa, o usuário mostra uma imagem com uma característica que ele esteja procurando, como por exemplo as cores predominantes ou um desenho com uma forma aproximada do objeto desejado. Esses sistemas geralmente são projetados e desenvolvidos por cientistas e usados exclusivamente por especialistas. Inicialmente são extraídas as características da imagem de consulta para que a seguir sejam comparadas com aquelas armazenadas na base de imagens. O sistema retorna então a imagem ou o conjunto de imagens mais similares ao padrão inicial de consulta, dependendo de como foi projetado o sistema (Figura 2.2).

Procurar um objeto específico em uma imagem é um grande desafio para os descritores globais. Uma possível solução para este problema é a técnica de dicionários visuais (*Visual Dictionary*), que divide a imagem em diferentes regiões, normalmente utilizando técnicas de aprendizado não supervisionado para agrupamento como K-médias, PCA (*Principal Component Analysis*) e modelo misturas de gaussianas (Titterington et al., 1985; Fernando et al., 2011). O vetor de característica de cada região torna-se uma "palavra" visual do dicionário. O objetivo é que diferentes regiões da imagem sejam associadas a diferentes "palavras", como vegetação, rochas, céu claro, nuvens, etc. Esta técnica é usada juntamente com uma outra de muito sucesso para recuperação de textos chamada de *Bag of Words* (BoW), que considera os documentos ou frases apenas pelo conjunto de palavras ignorando qualquer estrutura intrínseca. Depois de criado um dicionário com as palavras conhecidas, é criado um vetor onde cada um dos elementos representa uma palavra do dicionário. Cada frase (ou trecho de

texto) é representado pela quantidade de ocorrências de cada uma das palavras do dicionário, ou seja, um histograma que não preserva a ordem das palavras do texto. Assim, com os dicionários visuais é usada uma abordagem semelhante que em CBIR é também chamada de *Bag of Words* ou *Bag of Features*. A criação de um bom dicionário visual é o principal desafio ao se empregar essa técnica (Valle e Cord, 2009).

Também é comum alguns trabalhos utilizarem mais de um descritor para recuperação de imagens, representando diferentes características da imagem (como por exemplo cor e textura). Para isso, os descritores podem ser combinados pela aplicação de diferentes pesos a cada um dos descritores utilizados (Dorairaj e Namuduri, 2004; Rui et al., 1998; Vadivel et al., 2004). Outros trabalhos propõem a utilização de técnicas mais sofisticadas (Kherfi et al., 2004; Torres et al., 2009; Arevalillo-Herráez et al., 2010) para realizar essa combinação, como as analisadas nas Seções 3.3 e 3.4. Conforme apresentado no início deste Capítulo, a definição adotada nesta tese permite que diferentes características sejam codificadas em métricas e vetores diferentes na combinação de descritores.

Caso o resultado da busca não seja satisfatório, é possível avaliar qual é o descritor mais adequado, definir uma métrica diferente para cálculo de similaridade entre os vetores de característica ou ainda criar uma maneira de combinar diferentes descritores. Especialistas modificam então o sistema a fim de otimizá-lo para resolução do problema em questão. A técnica de busca por similaridade é normalmente utilizada em problemas específicos como reconhecimento de impressões digitais, placas de veículos, faces entre outros. Uma arquitetura típica de um sistema de recuperação de imagens por conteúdo é mostrada na Figura 2.3 (Torres e Falcão, 2006).

Para deduzir as características de um determinado problema, em muitos casos são utilizadas ferramentas para aprendizado de máquina (supervisionado ou não). No aprendizado supervisionado é apresentado um conjunto de treinamento, consistindo de imagens de entrada e suas imagens de saída desejadas. O objetivo dos métodos que utilizam este tipo de aprendizado (Zhang et al., 2001; Town e Sinclair, 2001; Luo e Savakis, 2001; Feng e Chua, 2003) é associar características de baixo nível à uma consulta realizada, aprendendo como utilizar os vetores de característica de forma a realizar a busca desejada. O aprendizado não supervisionado visa agrupar as imagens baseado nas distâncias dos vetores de característica, descrevendo como as imagens são organizadas sem a intervenção do usuário. Este tipo de aprendizado é normalmente utilizado por métodos (Shi e Malik, 2000; Lejsek et al., 2008; Valle et al., 2008) que visam possibilitar a busca de imagens em bases grandes. Esses agrupamentos de imagens geralmente são armazenados em estruturas de dados (Ciaccia et al., 1997; Traina Jr. et al., 2002; Vieira et al., 2010) para aumentar a eficiência na busca de imagens.

As buscas por similaridade podem ser úteis em diversos contextos, como verificar se um logotipo similar já foi registrado, buscar rostos suspeitos para prevenção de crime e detectar nudez em imagens ou vídeo. Um dos maiores desafios na área de recuperação de imagens é a criação de modelos

computacionais para responder questões como “mostre imagens de pessoas andando de bicicleta na praia”, buscando reproduzir a necessidade de um usuário comum. Como já citado, não existe ainda um sistema computacional que consiga imitar o funcionamento de recuperação e busca visual realizada pelo cérebro humano, ainda mais sem ter as mesmas informação adquiridas pelo usuário ao longo da sua existência. Por isso, diferentemente dos sistemas usados para resolver problemas de domínios específicos, sistemas de recuperação de imagens tentam resolver buscas utilizando-se da experiência do usuário, mimetizando assim nosso conhecimento para busca de similaridade.

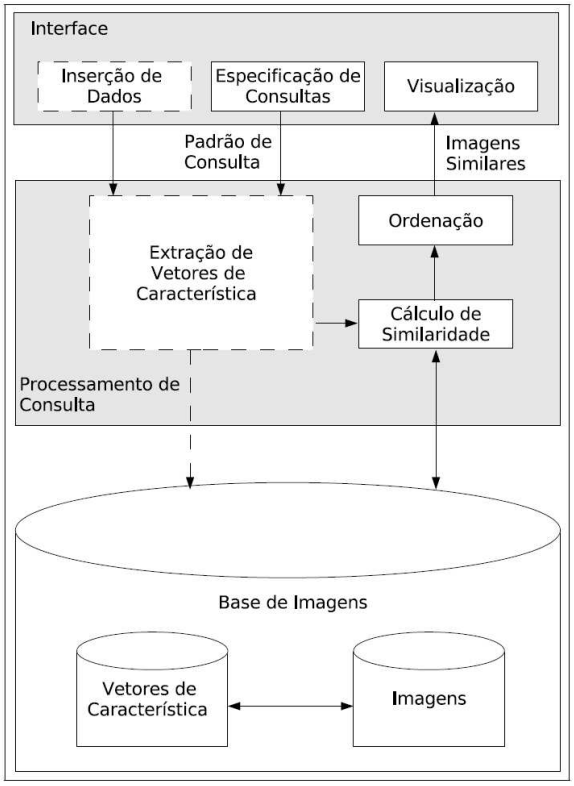


Fig. 2.3: Arquitetura de um sistema de recuperação de imagens por conteúdo.

Diferentemente dos métodos *off-line* para recuperação de informação, o processo de realimentação de relevância é uma técnica de aprendizado interativo e iterativo que visa deduzir a vontade do usuário em tempo condizente com a participação do usuário no processo de busca. A *realimentação de relevância* (RF - *Relevance Feedback*) é uma ferramenta poderosa originalmente usada para recuperação de documentos (Rocchio, 1971) e resgatada com sucesso para recuperação de imagens como nos sistemas MARS (Rui et al., 1997) e MindReader (Ishikawa et al., 1998). Essa técnica tem por objetivo possibilitar que o usuário expresse a sua necessidade sem recorrer a ajustes de propriedades de baixo nível (cor, textura e forma). Para isso, o usuário apenas indica durante algumas iterações quais

imagens retornadas pelo sistema são relevantes (em alguns casos também quais as irrelevantes). A informação de quais imagens interessam ou não ao usuário é utilizada para o aprendizado do sistema. A cada iteração, o algoritmo define quais são as propriedades visuais que melhor definem as imagens relevantes a partir das informações fornecidas pelo usuário. Ao final, o sistema retorna as imagens mais similares ao padrão desejado pelo usuário.

Esta técnica endereça duas questões referentes ao processo de recuperação de imagens por conteúdo. A primeira delas reside na lacuna semântica (*semantic gap*) entre as propriedades visuais de alto nível, através das quais o usuário tem a percepção da informação visual, e a descrição de baixo nível (vetores de característica) utilizada para a representação das imagens. A outra questão diz respeito ao caráter subjetivo da percepção da imagem pelo usuário. Diferentes pessoas (ou a mesma pessoa em diferentes circunstâncias) podem ter percepções visuais distintas de uma mesma imagem. Com a realimentação de relevância essas duas questões são tratadas de forma transparente para o usuário.

Essa técnica envolve os seguintes passos:

1. a seleção de uma imagem de exemplo pelo usuário que será apresentada inicialmente ao sistema;
2. a busca pelo sistema das imagens mais similares de acordo com as suas propriedades de mais baixo nível;
3. a indicação pelo usuário de quais imagens são relevantes ou não de acordo com a pesquisa realizada;
4. o aprendizado do desejo do usuário através de suas escolhas;
5. o retorno de imagens cada vez mais mais relevantes de acordo com a escolha do usuário.

Os passos 3 a 5 são repetidos até que o resultado esperado tenha sido alcançado, mas é altamente recomendado que esse processo termine em poucas iterações. Esta é a estratégia mais comum e nesta tese é denominada de gulosa (Seção 3.1). A inclusão do aprendizado na arquitetura apresentada na Figura 2.3 leva à criação da Figura 3.1.

Existem muitos estudos em cada uma das etapas desse processo, criando desde descritores mais eficientes a métodos para garantir a escalabilidade para grandes bases de imagens (Lejsek et al., 2008; Valle et al., 2008). O algoritmo de aprendizado (passo 4) é um ponto crucial para a definição de um mecanismo de realimentação de relevância. Em alguns trabalhos, o aprendizado consiste em estimar qual função de extração de característica ou a métrica de comparação que melhor represente o padrão

de consulta. Em outros, atribuem-se pesos a cada posição do vetor de característica ajustando os pesos nas características de baixo nível a fim do sistema adaptar-se às necessidades do usuário. No caso de se utilizar mais de um descritor, alguns sistemas buscam encontrar a melhor forma de combinar as diferentes informações acerca da imagem.

Já que os vetores de característica, e consequentemente as representações das imagens da base, podem ser interpretados como pontos no espaço, as informações das imagens marcadas pelo usuário durante as iterações podem ser utilizadas para mudar a representação definida pela imagem inicial. Na Figura 2.4a (QPM – *Query Point Movement*), o centro geométrico da pesquisa move-se de acordo com os exemplos positivos (imagens relevantes, por exemplo) de cada iteração. Outras técnicas utilizam múltiplos pontos de pesquisa (*multi-point query*), através dos exemplos positivos, para recuperar as imagens das próximas iterações. Dependendo da distância entre os múltiplos pontos de pesquisa são formadas diferentes regiões e formas no espaço de característica (Figura 2.4b). Os métodos baseados nesta técnica (Porkaew et al., 1999; Zhou et al., 2006; Liu et al., 2009; Su et al., 2011) geralmente usam a média entre as imagens relevantes ou os centros de agrupamentos (*clusters*) formados por elas para que o resultado da busca seja mais adequado ao desejo do usuário.

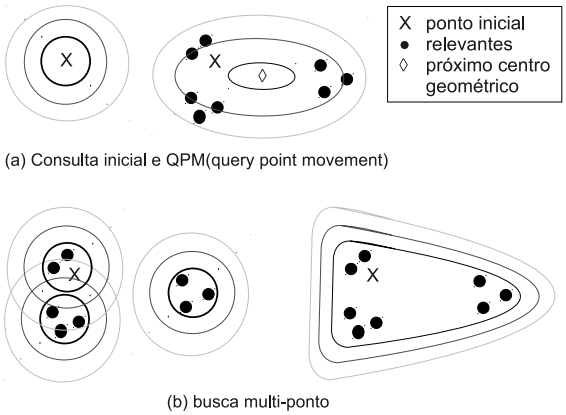


Fig. 2.4: Técnicas de realimentação de relevância por ajuste de peso em suas característica.

O modelo estatístico Bayesiano foi um dos primeiros classificadores a ser empregado (Cox et al., 2000; King e Jin, 2003; Giacinto e Roli, 2004; Xu e Akella, 2008) utilizando o usuário como tutor para classificar as imagens através de seus atributos de mais baixo nível (vetor de característica). Um classificador Bayesiano determina a probabilidade de que uma determinada característica esteja relacionada com a presença (ou ausência) de qualquer outra característica. Por exemplo, uma fruta pode ser considerada como uma maçã, se ela for redonda e vermelha. Mesmo que as características sejam interdependentes, elas são consideradas de maneira independente. Cada vetor de característica recebe um valor de probabilidade que é usado pelo classificador Bayesiano para determinar quais

são as imagens mais prováveis de satisfazer a opinião do usuário, dependendo das indicações feitas anteriormente por ele. Uma vantagem do classificador Bayesiano em relação aos baseados em pontos de pesquisa é que ele requer menor quantidade de dados de treinamento para estimar os parâmetros necessários para a classificação.

Alguns trabalhos utilizam técnicas de aprendizado de máquina como redes neurais artificiais para recuperação de imagens (Laaksonen et al., 2002; Srinivasa et al., 2006; Anh et al., 2010). Máquina de Vetor de Suporte (SVM – *Support Vector Machine*) é atualmente a técnica mais utilizada para este fim. O objetivo deste método é encontrar um hiperplano que separa duas classes distintas em um espaço multidimensional (Figura 2.5). Tong et al. (Tong e Chang, 2001) propuseram um método de realimentação de relevância usando SVM para separar imagens relevantes das não relevantes (irrelevantes). A cada iteração, são mostradas para o usuário as imagens mais próximas do hiperplano, ou seja, as mais ambíguas consideradas como sendo as mais representativas para o aprendizado. No final do processo (após a última iteração), as imagens mais afastadas do hiperplano no lado relevante são apresentadas para o usuário. Muitos trabalhos recentes (Philipp-Foliguet et al., 2009; Wu et al., 2010; Wang et al., 2011) utilizam SVM e buscam otimizar esse método para realimentação de relevância.

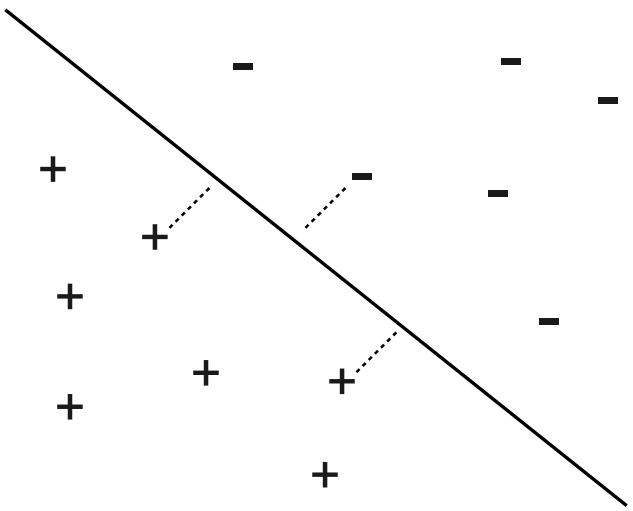


Fig. 2.5: Exemplo simples de separação de classes via SVM.

O método de realimentação de relevância proposto neste trabalho utiliza o classificador baseado em floresta de caminhos ótimos. A Seção a seguir apresenta uma breve explicação sobre o classificador OPF (Papa et al., 2009).

2.2.1 Classificador baseado em floresta de caminhos ótimos

O classificador OPF supervisionado originalmente concebido para reconhecimento de padrões (Papa et al., 2009) é pela primeira vez usado para a tarefa de recuperação de imagens por conteúdo (Silva et al., 2010). Esse classificador é modelado em uma floresta de caminhos ótimos, onde os nós são amostras (imagens no caso desta tese) e os arcos são ponderados pela distância entre as amostras. O conjunto de treinamento do classificador é um grafo completo $(\mathcal{T}, A)$, onde o conjunto $\mathcal{T}$ de nós contém as imagens rotuladas pelo usuário como relevantes ou não e A contém os arcos não direcionados ponderados por uma função de distância $d(s,t)$, entre as imagens s e t.

Calculando uma MST (*Minimum Spanning Tree* ou árvore espalhada mínima) (Cormen et al., 1990) em $(\mathcal{T}, A)$ é obtido um grafo simples conexo e acíclico cujos nós são todas as amostras de $\mathcal{T}$ e os arcos são não direcionados de modo que a soma dos pesos $d(s,t)$ das arestas é mínima (Figura 2.6). Esta árvore é ótima no sentido de que a soma dos pesos de seus arcos é mínima se comparada às outras árvores espalhadas no grafo completo.

Os elementos adjacentes na MST com diferentes rótulos em $\mathcal{T}$ são definidos como *protótipos*, isto é, elementos mais próximos entre relevantes e irrelevantes no caso desta tese. Removendo-se os arcos entre as diferentes classes, tais amostras adjacentes são armazenados nos conjuntos $\mathcal{S}_R$ e $\mathcal{S}_I$ de protótipos relevantes e irrelevantes, respectivamente (Figura 2.7). Formalmente, em cada arco (s,t) na MST, $\lambda(t)$ é a classe da imagem $t \in \mathcal{T}$ e $\lambda(s)$ é a classe da imagem $s \in \mathcal{T}$, que podem ser relevantes ou irrelevantes. Se $\lambda(s) \neq \lambda(t)$ então s e t são marcados como sendo protótipos. Se $\lambda(s)$ é relevante e $\lambda(t)$ é irrelevante, s é inserido em $\mathcal{S}_R$ e t é inserido em $\mathcal{S}_I$. Estes conjuntos são usados para calcular uma floresta de caminhos ótimos em $\mathcal{T}$.

Dado um grafo completo $(\mathcal{T}, A)$, um caminho π_t no grafo é uma sequência $\langle t_1, t_2, \ldots, t_n \rangle$ de nós que termina no nó $t_n = t$. Um caminho π_t é chamado de trivial se $\pi_t = \langle t \rangle$. A todo caminho π_t é associado um valor de custo definido pela função $c(\pi_t)$. Um caminho π_t é ótimo quando $c(\pi_t) \leq c(\pi_t{}')$ para qualquer caminho $\pi_t{}'$, onde π_t e $\pi_t{}'$ terminam no mesmo nó t. O custo do caminho entre o nó inicial $R(\pi_t) = t_1$ e o nó final $t_n = t$ é definido por:

$$c(\pi_t) \;=\; \max_{i=1,2,\ldots,n-1} \{d(t_i, t_{i+1})\} \tag{2.10}$$

e, o caminho mais fortemente conexo é inversamente proporcional ao peso máximo dos arcos:

$$C(t) \;=\; \min_{\forall \pi_t \in (\mathcal{T}, A)} \{c(\pi_t)\}. \tag{2.11}$$

O Algoritmo 1 (Papa et al., 2009) mostra o cálculo de uma floresta de caminhos ótimos para a função de custo C, que é uma extensão do algoritmo da IFT (*Image Foresting Transform*) (Falcão et al., 2004)

do domínio da imagem para o espaço de características. Para cada nó $t \in \mathcal{T}$, o algoritmo calcula o custo mínimo $C(t)$ e seu predecessor $P(t)$ do caminho ótimo cuja raíz é $R(t)$ ao nó terminal t. Também é calculada uma lista $\mathcal{T}'$ formada pelos nós do conjunto de treinamento ordenada pelo custo $C(t)$. Esta lista ordenada é usada para aumentar a eficiência na classificação.

Algoritmo 1: Algoritmo do classificador OPF

Entrada: Conjunto de treinamento $\mathcal{T}$, conjunto de protótipos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$, e um descritor (v, d).

Saída: Floresta de caminhos ótimos P, mapa de custo de caminhos C, mapa de raízes R e a lista do conjunto de treinamento ordenada $\mathcal{T}'$.

Auxiliares: Fila de prioridade Q e a variável de custo cst.

```
1  para todo s ∈ T\S_R ∪ S_I faça
2      C(s) ← +∞
3  fim
4  para todo s ∈ S_R ∪ S_I faça
5      C(s) ← 0, P(s) ← nil,
6      R(s) ← s e insira s em Q.
7  fim
8  enquanto Q não estiver vazia faça
9      Remova de Q uma amostra s com menor custo C(s) e insira s em T'.
10     para todo t ∈ T onde C(t) > C(s) faça
11         Calcule cst ← max{C(s), d(s,t)}.
12         se cst < C(t) então
13             se C(t) ≠ +∞ então
14                 remova t de Q.
15             fim
16             P(t) ← s, R(t) ← R(s) e C(t) ← cst.
17             Insira t em Q.
18         fim
19     fim
20 fim
```

As linhas de 1 a 7 inicializam os valores para os mapas de custo, predecessor e raíz, forçando os caminhos ótimos a iniciar em $\mathcal{S}_R \cup \mathcal{S}_I$ e inserir as raízes em Q. O laço externo calcula um caminho ótimo de $\mathcal{S}_R \cup \mathcal{S}_I$ para cada nó $s \in \mathcal{T}$ na ordem decrescente de custo (linhas 8 a 20). A cada iteração, é obtido um caminho com custo mínimo $C(s)$ em P. O último nó s é removido de Q e sua ordem

é preservada em $\mathcal{T}'$ (linha 9). As linhas 10 a 19 avaliam se o caminho que chega a um nó adjacente $t \neq s$, através de s, é melhor do que o caminho atual, atualizando Q, $C(t)$, $R(t)$ e $P(t)$ de acordo com o resultado.

Para o caso de realimentação de relevância, o objetivo é obter uma partição ótima de $\mathcal{T}$ na qual os conjuntos de imagens previamente rotuladas como relevantes e irrelevantes irão competir entre si para classificar as demais imagens da base $\mathcal{Z}\backslash\mathcal{T}$ de acordo com a floresta de caminhos ótimos. O caminho ótimo para cada imagem $t \in \mathcal{Z}\backslash\mathcal{T}$ pode ser facilmente identificado buscando qual dos nós de treinamento $s^* \in \mathcal{T}$ fornece o menor valor na Equação 2.12 (Figura 2.8).

$$C(t) = \min_{\forall s \in \mathcal{T}'}\{\max\{C(s), d(s,t)\}\}, t \in \mathcal{Z}\backslash\mathcal{T}, s \in \mathcal{T}' \tag{2.12}$$

O nó s^* é o predecessor $P(t)$ no caminho ótimo definido pelo nó terminal t, e portanto, a imagem t é classificada como pertencente à classe $\lambda(R(s^*))$, ou seja, ela recebe o mesmo rótulo da raiz da floresta na qual ela é mais fortemente conexa. A função de $\mathcal{T}'$ no algoritmo acima é aumentar a velocidade na estimativa da Equação 2.12, que pode parar quando $\max\{C(s), d(s,t)\} < C(p)$ para um nó p cuja posição em $\mathcal{T}'$ sucede a posição de s (Papa et al., 2010). Como $\mathcal{T}'$ está ordenado por custo, se $C(p)$ for maior do que $\max\{C(s), d(s,t)\}$ então os custos dos próximos elementos também serão maiores.

O Capítulo 3 apresenta os métodos propostos nesta tese e os resultados comparados aos obtidos por técnicas tradicionais são mostrados no Capítulo 4.

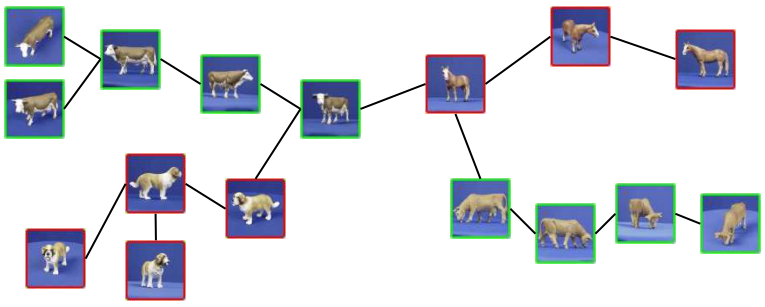


Fig. 2.6: Árvore espalhada mínima usando imagens da base ETH-80 (Leibe e Schiele, 2003).

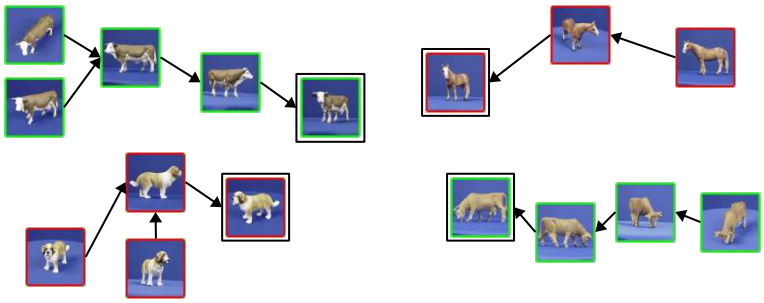


Fig. 2.7: Floresta de caminhos ótimos de imagens relevantes e irrelevantes.

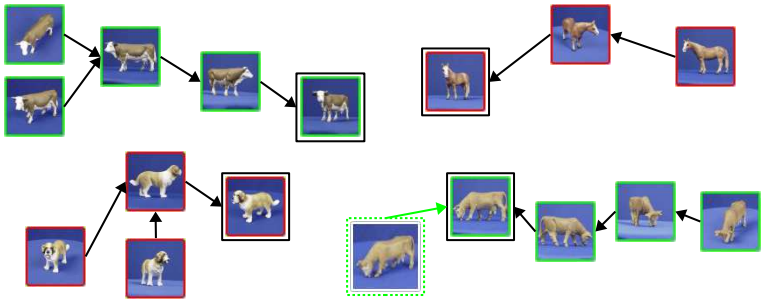


Fig. 2.8: Imagem $t \in \mathcal{Z} \backslash \mathcal{T}$ classificada como relevante.

Capítulo 3

Métodos de CBIR baseados em realimentação de relevância e floresta de caminhos ótimos

Este capítulo apresenta as contribuições desta tese na área de recuperação de imagem por conteúdo utilizando realimentação de relevância com os algoritmos $GOPF_{RF}$ (Realimentação de relevância usando OPF e aprendizado guloso) (Silva et al., 2010), $POPF_{RF}$ (Realimentação de relevância usando OPF e aprendizado planejado) (Silva et al., 2011), OPF_{MSPS} (Realimentação de relevância usando OPF e MSPS)[1], OPF_{GP} (Realimentação de relevância usando OPF e Programação Genética)[1] e $OPF_{Bi-Level}$ (Realimentação de relevância usando OPF e abordagem Bi-Level).

A Figura 3.1 estabelece, sob a nossa ótica, a arquitetura de um sistema de recuperação de imagens por conteúdo com realimentação de relevância. Primeiro, o usuário seleciona uma imagem de exemplo que será apresentada inicialmente ao sistema. A busca por similaridade (Figura 2.2) mostra as imagens mais próximas de acordo com alguma característica. O retângulo pontilhado mostra o processo de realimentação de relevância. Neste processo são exibidas as imagens para o usuário, ele indica quais imagens são relevantes ou não para a consulta e, de acordo com a indicação, é realizado o aprendizado e recuperação das próximas imagens a serem exibidas ao usuário.

O retângulo cinza da Figura 3.1 destaca o foco desta tese, que é a criação de um novo método para o processo de aprendizagem baseado em um classificador por floresta de caminhos ótimos. São também abordadas duas técnicas de combinação de descritores para serem utilizadas juntamente com o método de realimentação de relevância a fim de auxiliar e melhorar o processo de aprendizagem. A

[1] A.T. Silva, J.A. dos Santos, A.X. Falcão, R. da S. Torres e L.P. Magalhães. "Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning". Computer Vision and Image Understanding (submetido em dezembro de 2010).

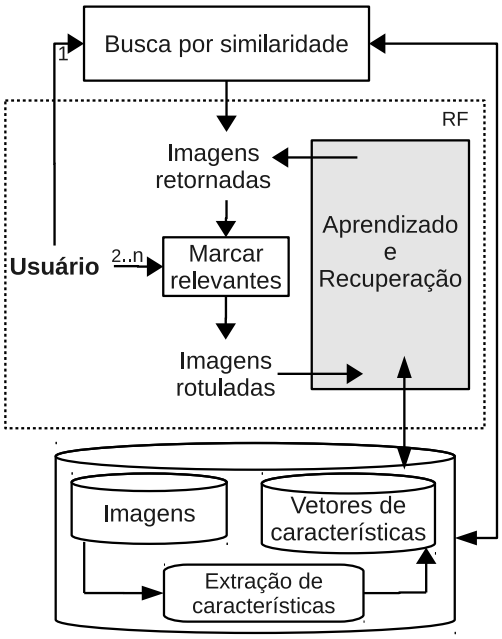


Fig. 3.1: Arquitetura de um sistema de recuperação de imagens por conteúdo com realimentação de relevância.

primeira utiliza um método de otimização chamado MSPS, utilizado pela primeira vez para combinação de descritores, enquanto a outra é uma técnica consolidada baseada em programação genética. Também é mostrado neste capítulo que o uso do classificador OPF pode ser utilizado para destacar regiões de interesse nas imagens da base, melhorando assim a eficácia do processo de recuperação de imagens.

Existem dois paradigmas de aprendizagem em realimentação de relevância em relação às imagens retornadas. No primeiro, a cada iteração tenta-se retornar sempre as imagens que o usuário considera mais relevantes, sendo o paradigma mais utilizado em CBIR. Nesta tese este paradigma é denominado como *guloso*. Em outros casos, como nos métodos baseados em SVM, o usuário estabelece em quantas iterações o sistema deverá aprender antes de retornar as imagens ordenadas por relevância. Durante estas iterações o sistema apresenta as imagens mais informativas para auxiliar a aprendizagem do sistema. Este último paradigma nesta tese é chamado de *planejado*. Métodos baseados em SVM por vezes são chamados de *Active Learning*, mas este termo é um conceito mais geral e apenas um de seus tipos utiliza imagens mais informativas. Além disso, nesses métodos as imagens mais relevantes são exibidas uma única vez ao final. No paradigma planejado existem duas iterações. Uma semelhante ao paradigma guloso que retorna as imagens mais relevantes e outra interna que retorna as imagens mais informativas para o processo de aprendizado.

Embora o número de iterações necessárias para o aprendizado pareça ser maior na abordagem pla-

nejada, isso não é necessariamente verdade, como é mostrado no Capítulo 4. Os algoritmos para cada um dos métodos desenvolvidos nesta tese são apresentados em maior detalhe a seguir. A Seção 3.1 descreve o treinamento do método de recuperação de imagens baseado em floresta de caminhos ótimos usando o paradigma guloso e a Seção 3.2 descreve o algoritmo do método usando o paradigma planejado. Também esta tese mostra que esta nova técnica de realimentação de relevância utilizando floresta de caminhos ótimos ($GOPF_{RF}$ ou $POPF_{RF}$) pode facilmente utilizar combinação de descritores para capturar as diferentes características da imagem. Na Seção 3.3 é apresentado um novo método de combinação de descritores usando a técnica de otimização MSPS para ajustar os parâmetros de uma função. Na Seção 3.4 é mostrada a integração do método de realimentação de relevância usando o classificador OPF com uma técnica de programação genética para criar funções de combinação de descritores. Também é apresentada uma nova abordagem para realimentação de relevância em dois níveis de interesse (pixel e imagem) utilizando o classificador OPF e denominada $OPF_{Bi-Level}$ na Seção 3.5.

3.1 Aprendizado guloso usando OPF – $GOPF_{RF}$

Seja $\mathcal{Z}$ uma base de imagens, onde cada imagem $t \in \mathcal{Z}$ é representada por um descritor (v, d) e a similaridade entre duas imagens $s, t \in \mathcal{Z}$ é medida pela função de distância $d(s, t)$. O objetivo final da busca de imagem por conteúdo é retornar uma lista $\mathcal{X}$ com as N imagens mais relevantes de $\mathcal{Z}$ de acordo com a opinião do usuário em relação a uma dada imagem de consulta inicial q. A abordagem mais simples é retornar as N imagens $t \in \mathcal{Z}$ mais próximas a q em ordem crescente de acordo com $d(q, t)$, conforme apresentado no início do Capítulo 2.

Entretanto, devido à limitação do descritor (v, d) para representar a expectativa do usuário, essa abordagem frequentemente apresenta uma lacuna semântica tal que a lista $\mathcal{X}$ contém não só imagens relevantes para o usuário, como também irrelevantes. A técnica de realimentação de relevância é usada para contornar essa lacuna semântica. O usuário indica quais imagens são relevantes (ou irrelevantes) em $\mathcal{X}$, formando um conjunto de treinamento rotulado $\mathcal{T}$ que ganha novos elementos a cada iteração da realimentação de relevância por $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$. O sistema aprende a vontade do usuário a fim de trazer imagens mais relevantes a cada iteração. Esse processo é repetido até que usuário esteja satisfeito.

Nos métodos a serem descritos, o conjunto $\mathcal{T}$ é usado para a criação de uma floresta de caminhos ótimos. Inicialmente são estimadas as imagens mais representativas (protótipos) considerando-se o desejo do usuário conforme a Seção 2.2.1, criando os subconjuntos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$ de protótipos relevantes e irrelevantes. Cada imagem $t \in \mathcal{Z} \backslash \mathcal{T}$ da base é então classificada como relevante ou irrelevante de acordo com as raízes da floresta que oferece a t o caminho ótimo no grafo. As imagens

classificadas como relevantes pela OPF formam o conjunto $\mathcal{Y}$ e somente as N imagens mais similares deste conjunto são selecionadas para serem ordenadas e retornadas na próxima iteração. As imagens que o classificador considera como irrelevantes são descartadas.

Para ordenar as imagens do conjunto $\mathcal{Y}$ e retornar as imagens mais relevantes de acordo com o desejo do usuário, é possível utilizar diversas métricas (ver Seção 2.1), como por exemplo, através do custo mínimo para as florestas de relevantes.

Uma possível métrica para ordenação das imagens é calculada pela Equação 3.1, onde $\min\{d(t, \mathcal{S}_R)\}$ é a distância de t ao protótipo relevante mais próximo e $\min\{d(t, \mathcal{S}_I)\}$ é a menor distância aos protótipos irrelevantes.

$$\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I) = \frac{\min\{d(t, \mathcal{S}_R)\}}{\min\{d(t, \mathcal{S}_R)\} + \min\{d(t, \mathcal{S}_I)\}} \tag{3.1}$$

Nesta tese foram testadas diferentes métricas e a distância média normalizada entre os conjuntos de protótipos relevantes e irrelevantes (Equação 3.2) foi a que gerou melhores resultados considerando as medidas de avaliação apresentadas no Capítulo 4.

$$\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I) = \frac{\bar{d}(t, \mathcal{S}_R)}{\bar{d}(t, \mathcal{S}_R) + \bar{d}(t, \mathcal{S}_I)}, onde \tag{3.2}$$

$$\bar{d}(t, \mathcal{S}_R) = \frac{1}{|\mathcal{S}_R|} \sum_{\forall s \in \mathcal{S}_R} d(s, t), \tag{3.3}$$

$$\bar{d}(t, \mathcal{S}_I) = \frac{1}{|\mathcal{S}_I|} \sum_{\forall s \in \mathcal{S}_I} d(s, t). \tag{3.4}$$

No paradigma guloso, o sistema retorna uma nova lista $\mathcal{X} \subset \mathcal{Y}$ com as N imagens mais próximas de acordo com a distância média normalizada $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ entre t e os dois conjuntos de protótipos. O processo repete durante algumas iterações até que o usuário esteja satisfeito e, ao final do processamento, são apresentadas todas as imagens relevantes conhecidas das iterações anteriores junto com a lista $\mathcal{X}$ em ordem crescente calculada por $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.

O Algoritmo 2 apresenta a técnica de realimentação de relevância usando floresta de caminhos ótimos (Silva et al., 2010) utilizando o paradigma de aprendizado guloso. Ele retorna uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens na ordem decrescente de relevância. Na linha 1, a lista de treinamento $\mathcal{T}$ é inicializada como um conjunto vazio. Para uma dada imagem de consulta inicial q e uma base de imagens $\mathcal{Z}$, o algoritmo retorna inicialmente uma lista $\mathcal{X}$ com as N imagens $t \in \mathcal{Z}$ mais próximas a q na ordem crescente de $d(q, t)$ (linha 2). O processo de aprendizado é executado no laço principal (linhas 3 a 13). Na linha 4, o usuário marca as imagens de $\mathcal{X}$ que ele considera como sendo relevantes e as demais são rotuladas como irrelevantes. Estas marcações feitas pelo usuário criam o conjunto de treinamento $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ onde cada uma das imagens $s \in \mathcal{T}$ recebe um rótulo de relevância $\lambda(s)$. As

imagens consideradas relevantes são inseridas na lista $\mathcal{R}$ (linha 5) que armazena as imagens relevantes conhecidas durante o processo de realimentação de relevância. O treinamento do classificador OPF é executado na linha 6 conforme o Algoritmo 1. O algoritmo retorna os conjuntos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$ de protótipos relevantes e irrelevantes, as listas com predecessor $P(s)$, custo $C(s)$ e raiz $R(s)$ para todas as imagens $s \in \mathcal{T}$ assim como uma lista $\mathcal{T}'$ ordenada pela ordem decrescente de custo $C(s)$ para aumentar a performance em eficiência do classificador. Cada uma das imagens $t \in \mathcal{Z}\backslash\mathcal{T}$ é classificada como relevante ou irrelevante pela OPF usando a Equação 2.12 (linha 9). Somente as imagens que forem classificadas como relevante são inseridas no conjunto $\mathcal{Y}$ (linha 10). Esta lista é usada para criar a lista $\mathcal{X}$ com as N imagens em ordem crescente de acordo com a distância média normalizada $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ entre t e o conjunto de protótipos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$. Ao final do processo, são retornadas as imagens rotuladas como relevantes pelo usuário durante o processo de realimentação de relevância (conjunto $\mathcal{R}$) mais N imagens candidatas a relevantes.

Algoritmo 2: Algoritmo de realimentação de relevância usando OPF e paradigma guloso ($GOPF_{RF}$).

Entrada: Uma imagem de consulta inicial q, um descritor (v, d), o número N de imagens retornadas por iteração e um banco de imagem $\mathcal{Z}$.
Saída: Uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens retornadas na ordem decrescente de relevância.
Auxiliares: Conjunto $\mathcal{T}$ de imagens de treinamento, mapas $(R, P, C, \mathcal{T}')$ geradas pelo classificador OPF, conjuntos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$ de protótipos, conjunto $\mathcal{Y} \subset \mathcal{Z}\backslash\mathcal{T}$ de imagens classificadas como relevante pela OPF e uma lista ordenada $\mathcal{X} \subset \mathcal{Y}$ de N imagens retornadas a cada iteração.

1 $\mathcal{R} \leftarrow \emptyset$ e $\mathcal{T} \leftarrow \emptyset$.
2 Crie uma lista $\mathcal{X}$ com N imagens mais próximas a $t \in \mathcal{Z}$ em ordem crescente de $d(q, t)$.
3 **enquanto** *o usuário não estiver satisfeito* **faça**
4 Usuário marca as imagens relevantes e irrelevantes, criando um conjunto de treinamento $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ onde cada imagem $s \in \mathcal{T}$ recebe um rótulo de relevância $\lambda(s)$.
5 Insira as imagens relevantes de $\mathcal{X}$ no conjunto $\mathcal{R}$.
6 Calcule os conjuntos $\mathcal{S}_R$ e $\mathcal{S}_I$ de protótipos em $\mathcal{T}$ (Seção 2.2.1) e execute $(R, P, C, \mathcal{T}') \leftarrow OPF(\mathcal{T}, \mathcal{S}_R, \mathcal{S}_I, v, d)$ (Algoritmo 1).
7 $\mathcal{Y} \leftarrow \emptyset$.
8 **para** *toda imagem* $t \in \mathcal{Z}\backslash\mathcal{T}$ **faça**
9 Classifique t usando $(R, P, C, \mathcal{T}')$ e $\lambda(s)$ para $s \in \mathcal{S}_R \cup \mathcal{S}_I$ (Equação 2.12).
10 **se** $\lambda(t)$ *é relevante* **então** Insira t em $\mathcal{Y}$.
11 **fim**
12 Crie uma lista $\mathcal{X}$ com as N imagens mais próximas a $t \in \mathcal{Y}$ na ordem crescente de $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.
13 **fim**
14 Retorne $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

Os resultados dos experimentos realizados utilizando $GOPF_{RF}$ são apresentados na Seção 4.3 comparados aos resultados obtidos pelos métodos considerados como sendo o estado-da-arte em realimentação de relevância.

3.2 Aprendizado planejado usando OPF – $POPF_{RF}$

A diferença básica do paradigma planejado para o guloso é que o usuário informa durante o aprendizado em quantas iterações o sistema deverá aprender antes de apresentar as imagens mais relevantes. Isto é espelhado no algoritmo através de um laço interno de I iterações, que otimiza o aprendizado retornando em $\mathcal{X}$ as N imagens mais informativas a cada iteração. As imagens são apresentadas em ordem crescente da diferença absoluta entre os custos em relação a $\mathcal{S}_R$ e a $\mathcal{S}_I$ (Equação 3.5). Somente após a I-ésima iteração é que as imagens finalmente serão ordenadas pela distância média normalizada $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$. O objetivo do laço interno é apresentar as imagens que melhor auxiliam a classificar as imagens $t \in \mathcal{Z}\backslash\mathcal{T}$ como relevantes ou não. Para isto, o sistema busca neste laço interno as imagens mais informativas para o classificador OPF.

O Algoritmo 3 apresenta a técnica de realimentação de relevância usando floresta de caminhos ótimos utilizando o paradigma de aprendizado planejado ($POPF_{RF}$). Ele também retorna ao final uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens na ordem decrescente de relevância. As linhas 1 e 2 são as mesmas do Algoritmo 2 (guloso) e o processo de aprendizagem é executado dentro de um laço principal que vai da linha 3 até a linha 13.

Este algoritmo contém um laço interno nas linhas 5 a 11 que força a execução de I iterações, informadas pelo usuário na linha 4. As linhas 6 a 9 do algoritmo são essencialmente as mesmas do algoritmo anterior, que criam a lista $\mathcal{Y} \subset \mathcal{Z}\backslash\mathcal{T}$ de imagens candidatas a relevante. Entretanto, na linha 10 é criada uma lista $\mathcal{X}$ com as N imagens *mais informativas* entre as candidatas a relevante em $\mathcal{Y}$. As imagens mais informativas são aquelas que são mais prováveis de se tornarem protótipos em $\mathcal{S}_R \cup \mathcal{S}_I$ e são utilizadas para acelerar o processo de aprendizado. Elas são as imagens mais difíceis de classificar, já que estão na fronteira entre os conjuntos relevantes e irrelevantes. Assim, as imagens mais informativas são escolhidas entre as imagens $t \in \mathcal{Y}$ com menor diferença $d_c(t, \mathcal{S}_R, \mathcal{S}_I)$ entre seus valores de custo para os conjuntos de protótipos relevantes e irrelevantes. O conjunto $\mathcal{X}$, que é apresentado ao usuário a cada iteração, é então criado pela ordem crescente de $d_c(t, \mathcal{S}_R, \mathcal{S}_I)$, calculado pela Equação 3.5.

$$d_c(t, \mathcal{S}_R, \mathcal{S}_I) \quad = \quad |C_R(t) - C_I(t)|, \tag{3.5}$$

onde $C_R(t)$ é o custo do melhor caminho com raiz em $\mathcal{S}_R$ e $C_I(t)$ é o custo do melhor caminho com

raiz em $\mathcal{S}_I$. Pelo menos um deles é o caminho ótimo com nó terminal em t. Devido à distribuição dos dados testados, é utilizado somente o custo $C_R(t)$, já que em $\mathcal{Y}$ são armazenadas somente as imagens candidatas a relevantes de acordo com a linha 9 do algoritmo.

Algoritmo 3: Algoritmo de realimentação de relevância usando OPF e paradigma planejado ($POPF_{RF}$).

Entrada: Uma imagem de consulta inicial q, um descritor (v, d), o número N de imagens retornadas por iteração e um banco de imagem $\mathcal{Z}$.

Saída: Uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens retornadas na ordem decrescente de relevância.

Auxiliares: Conjunto $\mathcal{T}$ de imagens de treinamento, mapas $(R, P, C, \mathcal{T}')$ geradas pelo classificador OPF, conjuntos $\mathcal{S}_R \subset \mathcal{T}$ e $\mathcal{S}_I \subset \mathcal{T}$ de protótipos, conjunto $\mathcal{Y} \subset \mathcal{Z}\backslash\mathcal{T}$ de imagens classificadas como relevante pela OPF e uma lista ordenada $\mathcal{X} \subset \mathcal{Y}$ de N imagens retornadas a cada iteração.

1 $\mathcal{R} \leftarrow \emptyset$ e $\mathcal{T} \leftarrow \emptyset$.
2 Crie uma lista $\mathcal{X}$ com N imagens mais próximas a $t \in \mathcal{Z}$ em ordem crescente de $d(q, t)$.
3 **enquanto** *o usuário não estiver satisfeito* **faça**
4 Pergunte ao usuário pelo número I de iterações planejadas.
5 **para** $i = 1$ *até* I **faça**
6 Usuário marca as imagens relevantes e irrelevantes, criando um conjunto de treinamento $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ onde cada imagem $s \in \mathcal{T}$ recebe um rótulo de relevância $\lambda(s)$.
7 Insira as imagens relevantes de $\mathcal{X}$ no conjunto $\mathcal{R}$.
8 Calcule os conjuntos $\mathcal{S}_R$ e $\mathcal{S}_I$ de protótipos em $\mathcal{T}$ (Seção 2.2.1) e execute $(R, P, C, \mathcal{T}') \leftarrow OPF(\mathcal{T}, \mathcal{S}_R, \mathcal{S}_I, v, d)$ (Algoritmo 1).
9 Classifique toda imagem $t \in \mathcal{Z}\backslash\mathcal{T}$ usando $(R, P, C, \mathcal{T}')$ e $\lambda(s)$ para $s \in \mathcal{S}_R \cup \mathcal{S}_I$ (Equação 2.12) e crie o conjunto $\mathcal{Y}$ com as imagens candidatas a relevante.
10 Crie uma lista $\mathcal{X}$ com as N imagens mais próximas a $t \in \mathcal{Y}$ na ordem crescente de $d_c(t, \mathcal{S}_R, \mathcal{S}_I)$ (Equation 3.5).
11 **fim**
12 Crie uma lista $\mathcal{X}$ com as N imagens mais próximas a $t \in \mathcal{Y}$ na ordem crescente de $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.
13 **fim**
14 Retorne $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

A lista ordenada $\mathcal{T}'$ é usada para obter $C_R(t)$ e $C_I(t)$ comparando a busca pelos nós predecessores $s \in \mathcal{T}'$ tendo sua raiz $R(s)$ em $\mathcal{S}_R$ ou em $\mathcal{S}_I$.

$$C_R(t) = \min_{\forall s \in \mathcal{T}' | R(s) \in \mathcal{S}_R} \{\max\{C(s), d(s,t)\}\} \quad (3.6)$$

$$C_I(t) = \min_{\forall s \in \mathcal{T}' | R(s) \in \mathcal{S}_I} \{\max\{C(s), d(s,t)\}\}. \quad (3.7)$$

Após a I-ésima iteração, uma lista $\mathcal{X}$ é finalmente apresentada ao usuário com as N imagens $t \in \mathcal{Y}$ candidatas a relevante ordenadas por $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ (linha 12). Todo o processo é repetido enquanto o usuário não estiver satisfeito. Após o término do processo, as imagens no conjunto $\mathcal{R}$ e as últimas imagens candidatas a relevante em $\mathcal{X}$ são retornadas em ordem crescente de $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ (linha 14).

O conjunto $\mathcal{Y}$ criado na linha 9 poderia incluir também as imagens consideradas irrelevantes, já que o objetivo é retornar as imagens mais difíceis de classificar que estão igualmente próximas a $\mathcal{S}_R$ e $\mathcal{S}_I$. Testes realizados nesta tese indicaram que a inclusão de imagens consideradas irrelevantes não ajuda a melhorar a eficiência do aprendizado. Para os testes realizados, a razão entre as imagens relevantes e o total de imagens da base é pequena e o número de falsos positivos tende a ser muito maior do que o número de falsos negativos. Ou seja, as imagens classificadas como irrelevantes provavelmente o são, sendo mais fácil encontrar imagens erroneamente classificadas como sendo relevantes.

Os resultados dos experimentos realizados utilizando $POPF_{RF}$ são apresentados na Seção 4.3 comparados aos resultados obtidos pelos métodos considerados como sendo o estado-da-arte em realimentação de relevância.

3.3 Descritor composto usando MSPS

Os métodos $GOPF_{RF}$ e $POPF_{RF}$ apresentados utilizam um único descritor para o aprendizado. Esta Seção descreve como usar diferentes características da imagem juntamente com o método de realimentação de relevância baseado em floresta de caminhos ótimos.

Devido à limitação de um único descritor (v, d) para representar o desejo de um usuário, vários trabalhos (Dorairaj e Namuduri, 2004; Rui et al., 1998; Vadivel et al., 2004; Cox et al., 2000; Kherfi et al., 2004; Torres et al., 2009; Arevalillo-Herráez et al., 2010) propõem a utilização de combinação de descritores para reduzir essa limitação. A distinção entre extração de características e função de similaridade é importante ao se combinar descritores que representam propriedades distintas (cor e forma, por exemplo), como foi demonstrado por Torres et al. Torres et al. (2009). O método apresentado nesta seção utiliza o conceito de descritor composto definido neste trabalho, no qual diferentes funções de similaridade podem ser utilizadas em conjunto.

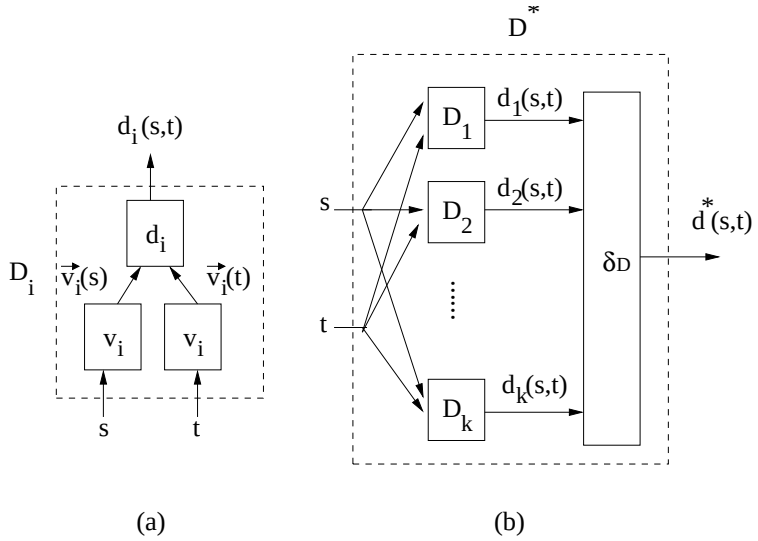


Fig. 3.2: Descritor (a) simples e (b) composto $D^* = (\mathcal{D}, \delta D)$.

Neste caso, cada imagem da base $\mathcal{Z}$ é representada por um conjunto de descritores D_i, $i = 1, 2, \ldots, n$ e cada descritor D_i é um par (v_i, d_i). Um descritor composto D^* é uma tupla $(\mathcal{D}, \delta D)$, onde $\mathcal{D}$ é um conjunto $\{D_1, D_2, \ldots, D_n\}$ de descritores individuais e δD é a função que combina os valores de distância calculados pelos descritores individuais e gera um valor de distância final $d^*(s,t)$ (ver Figura 3.2). Para utilizar o conceito de descritor composto nas técnicas de realimentação de relevância apresentadas nesta tese, $d^*(s,t)$ é usado para definir os pesos dos arcos do grafo completo definido por $\mathcal{T}$ no Algoritmo 1 e para calcular as distâncias $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ e $d_c(t, \mathcal{S}_R, \mathcal{S}_I)$ (respectivamente Equações 3.2 e 3.5).

Diferentes descritores normalmente têm métricas e valores finais em escalas de valores diferentes. Por isso, é importante que todos os valores de similaridade sejam normalizados ao se fazer esta combinação. Nos resultados apresentados no Capítulo 4 foi utilizada a normalização Gaussiana (Rui et al., 1998).

A função δD pode ser definida de diferentes formas e os resultados mostrados na Seção 4.4 utilizam a seguinte equação:

$$\delta D(s,t) \;\; = \;\; \sum_{i=1}^{n} d_i^{\theta_i}(s,t), \tag{3.8}$$

onde $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$ é o conjunto de parâmetros da função de combinação, tal que $0 \leq \theta_i \leq 2$. Esta equação define um peso não linear para cada um dos descritores escolhidos para compor a função de combinação. O valor 0 indica que os valores de distância definidos pelo descritor não afetam a combinação. Quanto maior o valor de θ_i maior a influência do descritor na combinação, sendo essa

influência não linear.

A melhor função de combinação é aquela que traz primeiramente as imagens mais relevantes para uma determinada consulta. A lista $\mathcal{T}$ com as imagens rotuladas previamente pelo usuário (relevantes/irrelevantes) é usada para encontrar esta função de combinação. Como estimativa inicial, seria possível utilizar a imagem de consulta q e encontrar a função de combinação que traz as imagens mais relevantes no início ao ordenar as imagens $t \in \mathcal{T}$ utilizando $d^*(q,t)$ definida pela a função de distância δD (candidata à função de combinação). Entretanto, a estimativa da melhor função de combinação utilizada nesta tese utiliza não só a imagem de consulta q, mas todas as imagens rotuladas pelo usuário como relevantes. Esta abordagem é comprovadamente mais eficaz e robusta do que o método mais simples usando apenas a consulta inicial (Torres et al., 2009).

Seja $\mathcal{T}_R$ o subconjunto de imagens relevantes de $\mathcal{T}$, $\mathcal{T}_u$ é o conjunto de imagens de treinamento $t \in \mathcal{T}$ ordenadas pela distância $\delta D(t,u)$, e consequentemente $\boldsymbol{\theta}$, para cada uma das imagens relevantes $u \in \mathcal{T}_R$. Assim, para cada imagem relevante do conjunto de treinamento $u \in \mathcal{T}$ é criada uma lista ordenada de imagens $\mathcal{T}_u$. Existem diversas funções critério para avaliar uma função de combinação e os resultados apresentados no Capítulo 4 utilizam a função FFP4 (Torres et al., 2009) definida por

$$F(\mathcal{T},\mathcal{D},\delta D) \;=\; \frac{1}{\mid \mathcal{T}_R \mid} \sum_{\forall u \in \mathcal{T}_R} \sum_{k=1}^{|\mathcal{T}_u|} 7\lambda_k 0.982^k, \tag{3.9}$$

onde $\lambda_k \in \{0,1\}$ armazena o rótulo dado pelo usuário a cada posição k de $\mathcal{T}_u$, podendo ser relevante ($\lambda_k = 1$) ou irrelevante ($\lambda_k = 0$).

Uma nova abordagem para combinação de descritores foi desenvolvida nesta tese para determinar a importância de cada uma das diferentes características da imagem. É utilizado o método MSPS para determinar os parâmetros mais adequados da Equação 3.8 a fim de combinar os diferentes valores de similaridade obtidos de cada descritor. O objetivo é obter uma função de combinação apropriada e desta maneira aumentar ainda mais a eficácia da técnica de realimentação de relevância baseada em floresta de caminhos ótimos.

A partir de um estado inicial $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$, a ideia é encontrar o melhor vetor de deslocamento $\boldsymbol{\Delta}^*$ e atualizar o vetor de parâmetros para o próximo valor $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\Delta}^*$, repetindo este processo até encontrar um máximo da função de avaliação $F(\mathcal{T},\mathcal{D},\delta D)$, que por motivo de simplificação é definida como $F(\boldsymbol{\theta})$ já que são alterados apenas os valores de $\boldsymbol{\theta}$ em δD (Equação 3.8). Para o caso de combinação de descritores, todos os valores θ_i iniciam com valor 1, indicando que inicialmente todos os descritores têm a mesma importância na função de combinação.

A fim de tentar evitar máximos locais, o método perturba $\boldsymbol{\theta}$ em cada um dos n parâmetros θ_i e em diferentes escalas de deslocamento $j = 1, 2, \ldots, m$. A cada iteração, é estimado o valor de $F(\boldsymbol{\theta} + \boldsymbol{\Delta})$ para cada vetor de deslocamento $\boldsymbol{\Delta}$ resultante de todas as perturbações em cada eixo i e o

vetor resultante de todas as escalas m, como descrito a seguir.

$0 \leq \Delta_{i,j} \leq 1$ é um deslocamento positivo no parâmetro de índice i para a escala j. O método calcula a função F considerando:

- a melhor orientação da perturbação ao longo de cada parâmetro i, como $\boldsymbol{\Delta}^*_{i,j} = (0, \ldots, \Delta^*_{i,j}, \ldots, 0)$ para $\Delta^*_{i,j} \in \{\Delta_{i,j}, 0, -\Delta_{i,j}\}$, tal que

$$F(\boldsymbol{\theta} + \boldsymbol{\Delta}^*_{i,j}) \quad = \quad \max \left\{ \begin{array}{l} F(\boldsymbol{\theta} + \boldsymbol{\Delta}_{i,j}), \\ F(\boldsymbol{\theta}), \\ F(\boldsymbol{\theta} - \boldsymbol{\Delta}_{i,j}) \end{array} \right\} \tag{3.10}$$

- e o vetor resultante $\boldsymbol{\Delta}\mathbf{s}_j = \sum_{i=1}^{n} \boldsymbol{\Delta}^*_{i,j}$, $j = 1, 2, \ldots, m$.

Assim, a escolha de $\boldsymbol{\Delta}^*$ pode ser expressa por:

$$F(\boldsymbol{\theta} + \boldsymbol{\Delta}^*) \quad = \quad \max \left\{ \begin{array}{ll} F(\boldsymbol{\theta} + \boldsymbol{\Delta}^*_{i,j}) & i\text{=1,2,...,}n \text{ e } j\text{=1,2,...,}m. \\ F(\boldsymbol{\theta} + \boldsymbol{\Delta}\mathbf{s}_j) & j\text{=1,2,...,}m. \end{array} \right\} \tag{3.11}$$

O Algoritmo 4 ilustra o cálculo do melhor conjunto de parâmetros $\boldsymbol{\theta}$ para a Função 3.8 usando MSPS. Primeiro são calculados os valores para $\Delta_{i,j}$ de acordo com a descrição acima a fim de servir como entrada para o algoritmo juntamente com as escalas definidas previamente. A linha 1 inicializa o vetor $\boldsymbol{\theta}$ com valor 1 para todas as posições, dando inicialmente o mesmo peso para cada um dos descritores. Nas linhas 4 a 16 é procurado pelo deslocamento $\boldsymbol{\theta}^*$ que gera o valor máximo para a função critério. Se existe um deslocamento $\boldsymbol{\Delta}$ (definido na linha 8) que maximiza a função critério, o valor é armazenado em $\boldsymbol{\Delta}^*$, testando tanto deslocamentos positivos quanto negativos para as escalas (linhas 9–11). Se foi encontrado um vetor de deslocamento melhor, $\boldsymbol{\Delta}^*$ e V^* são atualizados na linha 12. Se a combinação dos deslocamentos $\boldsymbol{\Delta}\mathbf{s}$ gerar uma função de combinação melhor, $\boldsymbol{\Delta}^*$ e V^* são novamente atualizados nas linhas 15 e 16. O algoritmo para quando nenhuma outra combinação de parâmetros gera um resultado melhor que $\boldsymbol{\theta}$ para a função critério. Ao final é retornado na linha 19 o conjunto de parâmetros $\boldsymbol{\theta}$ a ser utilizado na Equação 3.8 para definir um descritor composto para ser utilizado nos métodos $GOPF_{RF}$ e $POPF_{RF}$, determinando o valor de distância $d^*(s,t)$ entre duas imagens $s, t \in \mathcal{Z}$.

Algoritmo 4: Algoritmo do método de otimização MSPS

Entrada: Deslocamentos Δ, número m de escalas e n de parâmetros (número de descritores).
Saída: Vetor $\boldsymbol{\theta}$ com os parâmetros que geram o maior valor para a função F.
Auxiliares: Vetores $\boldsymbol{\Delta}^*$, $\boldsymbol{\Delta}\mathbf{s}$ e $\boldsymbol{\theta}^*$ e valores i, j, V_0, V^-, V^+ e V^*.

$\boldsymbol{\theta} \leftarrow (1, \ldots, 1)$;
$V^* \leftarrow F(\boldsymbol{\theta})$ e $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$;
enquanto $V^* > V_0$ **faça**
    $V_0 \leftarrow V^*$ e $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$;
    para *j=1 to m* **faça**
        $\boldsymbol{\Delta}\mathbf{s} \leftarrow (0, \ldots, 0)$;
        para *i=1 to n* **faça**
            $\boldsymbol{\Delta} \leftarrow (0, \ldots, \Delta_{i,j}, \ldots, 0)$, $V \leftarrow V_0$, e $\boldsymbol{\Delta}^* \leftarrow (0, \ldots, 0)$;
            $V^+ \leftarrow F(\boldsymbol{\theta} + \boldsymbol{\Delta})$ e $V^- \leftarrow F(\boldsymbol{\theta} - \boldsymbol{\Delta})$;
            se $V^+ > V$ **então** $V \leftarrow V^+$ e $\boldsymbol{\Delta}^* \leftarrow \boldsymbol{\Delta}$;
            se $V^- > V$ **então** $V \leftarrow V^-$ e $\boldsymbol{\Delta}^* \leftarrow -\boldsymbol{\Delta}$;
            se $V > V^*$ **então** $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} + \boldsymbol{\Delta}^*$ e $V^* \leftarrow V$;
            $\boldsymbol{\Delta}\mathbf{s} \leftarrow \boldsymbol{\Delta}\mathbf{s} + \boldsymbol{\Delta}^*$;
        fim
        $V \leftarrow F(\boldsymbol{\theta} + \boldsymbol{\Delta}\mathbf{s})$;
        se $V > V^*$ **então** $V^* \leftarrow V$ e $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} + \boldsymbol{\Delta}\mathbf{s}$;
    fim
fim
Retorne $\boldsymbol{\theta}$.

Os resultados dos experimentos realizados utilizando OPF_{MSPS} são apresentados na Seção 4.4 comparando-os com os obtidos usando um único descritor e os obtidos com execução do método GP^+ (Ferreira et al., 2011), que pode ser considerado como estado-da-arte em combinação de descritores.

3.4 Descritor composto usando Programação Genética

O método MSPS procura por um conjunto de parâmetros ótimo para obter uma função de combinação. A abordagem apresentada nesta seção gera uma função de combinação usando programação genética (GP – *Genetic Programming*), que é uma técnica evolutiva de solução de problemas (Koza, 1992). Nesta abordagem o método GP é integrado a técnica de realimentação de relevância apresentada na Seção 3.1. Para isto, cada indivíduo GP representa uma função candidata δD, ou seja,

um descritor composto. O indivíduo é definido por uma estrutura de árvore, conforme ilustrado no exemplo da Figura 3.3. Os nós das folhas recebem os valores dados pelos diferentes descritores e os nós internos definem operações matemáticas. A Figura 3.3 utiliza três descritores e um conjunto de operações $\{+, -, /, sqrt\}$ nos nós internos, representando a Equação 3.12.

$$\delta D(t,s) \quad = \quad \frac{d_1(t,s) + d_3(t,s)}{d_2(t,s)} - \sqrt{d_2(t,s) + d_3(t,s)} \tag{3.12}$$

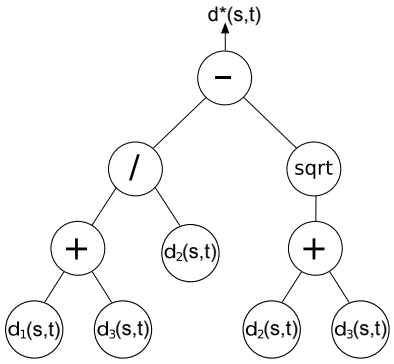


Fig. 3.3: Um descritor composto representado por uma árvore.

A programação genética procura funções de combinação evoluindo uma população $\mathcal{P}$ de n_p indivíduos por iteração durante g_n gerações. O melhor indivíduo da população gerada pela aplicação de transformações genéticas, como reprodução, mutação e *crossover* (Koza, 1992) é definido como função de combinação de descritores. A operação de reprodução seleciona os melhores indivíduos e os copia na próxima geração. A mutação é definida como uma manipulação aleatória que atua em apenas um indivíduo, selecionando um nó e substituindo a subárvore por outra gerada aleatoriamente. A operação de *crossover* combina o material genético de dois indivíduos trocando uma de suas subárvores.

Para uma dada lista $\mathcal{T}$ de imagens rotuladas (relevante/irrelevante) previamente por um usuário, o método tenta encontrar uma função de combinação δD. O Algoritmo 5 ilustra o método baseado em programação genética para encontrar uma função candidata. Uma população $\mathcal{P}$ inicia com n_p indivíduos gerados aleatoriamente (linha 1). Esta população evolui durante as g_n gerações através das operações genéticas (linhas 2 a 9). A função critério $F(\mathcal{T}, \mathcal{D}, \delta D_i)$ (Equação 3.9) é calculada para atribuir o valor de *fitness* para cada um dos indivíduos (linha 5). Esse valor é utilizado para selecionar os melhores indivíduos e armazená-los no conjunto $\mathcal{A}$. A seguir, as operações genéticas são aplicadas na população a fim de criar melhores indivíduos a cada geração (linhas 7 e 8). O último passo consiste em determinar qual é melhor indivíduo (maior valor de *fitness*) que será utilizado para criar a função de combinação utilizada para definir os pesos dos arcos do grafo completo definido por

$\mathcal{T}$ no Algoritmo 1 assim como para calcular as distâncias $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ e $d_c(t, \mathcal{S}_R, \mathcal{S}_I)$.

Algoritmo 5: GP Algorithm

Entrada: Conjunto de treinamento $\mathcal{T}$, conjunto $\mathcal{D}$ de descritores, tamanho n_p da população e o número g_n de gerações, percentuais de reprodução, mutação e *crossover*.

Saída: O melhor indivíduo δD^* (uma estrutura de árvore).

Auxiliares: População P de n_p indivíduos δD_i, um conjunto $\mathcal{A}$ de pares $(\delta D_i, F(\mathcal{T}, \mathcal{D}, \delta D_i))$, variáveis i e g.

1 $\mathcal{P} \leftarrow$ população inicial aleatória de n_p indivíduos;
2 $\mathcal{A} \leftarrow \emptyset$;
3 **para** *toda geração* $g = 1, 2, \ldots, n_g$ **faça**
4 **para** *todo indivíduo* $\delta D_i \in \mathcal{P}$, $i = 1, 2, \ldots, n_p$ **faça**
5 Insira $\mathcal{A} \leftarrow \mathcal{A} \cup (\delta D_i, F(\mathcal{T}, \mathcal{D}, \delta D_i))$;
6 **fim**
7 Ordene $\mathcal{A}$ em ordem crescente de $F(\mathcal{T}, \mathcal{D}, \delta D_i)$;
8 Crie uma nova população $\mathcal{P}$ de tamanho n_p aplicando as operações de reprodução, mutação e *crossover* aos melhores indivíduos em $\mathcal{A}$;
9 **fim**
10 Retorne o melhor indivíduo δD^* em $\mathcal{A}$ com o maior valor de $F(\mathcal{T}, \mathcal{D}, \delta D^*)$.

Os resultados dos experimentos realizados utilizando OPF_{GP} são apresentados na Seção 4.4 comparando-os com os obtidos usando um único descritor e com os obtidos com a execução do método GP^+, que pode ser considerado como estado-da-arte em combinação de descritores.

3.5 $OPF_{Bi-Level}$ – Aprendizado em dois níveis de interesse por realimentação de relevância baseada em OPF

O objeto de interesse em uma busca de imagens nem sempre ocupa toda a área de uma figura, sendo somente parte dela importante para o usuário. Imagens de pássaros, por exemplo, muitas vezes tem o fundo azul do céu, em virtude de serem fotografias de aves voando, ou com o fundo esverdeado se de aves em galhos de árvores. No caso dos descritores de cor, por exemplo, esse fundo irá interferir no resultado da busca, pois é sabido que características extraídas da imagem inteira não representam necessariamente as características dos diversos objetos contidos nela. Por esse motivo, muitos sistemas localizam objetos em diferentes regiões da imagem (Huang et al., 2010), utilizando métodos que particionam a imagem em regiões fixas (Ma e Manjunath, 1999) ou subtraem o fundo

Fig. 3.4: Seleção da região de interesse. A região preta é a área descartada.

da imagem para isolar a região de interesse onde os descritores serão calculados (Carson et al., 1999; Lu e Guo, 1999; Wang et al., 2001; Kim et al., 2003; Jing et al., 2004; Philipp-Foliguet et al., 2009).

Dentre suas contribuições esta tese apresenta um método semi-automático para busca de objetos por nível de interesse baseado no classificador OPF. Diferentemente de outros métodos, o usuário seleciona interativamente os objetos de interesse durante as iterações do processo de realimentação de relevância, descartando o que não lhe é útil. Ao marcar pixels (ver Figura 3.4), rotulando-os como objeto (linha azul) ou fundo (linha vermelha), o usuário define um conjunto de treinamento para criação de um classificador OPF, utilizando os valores de cor dos pixels e do rótulo que definem se o pixel pertence à classe de objeto e ou de fundo. Cada pixel marcado é representado por um vetor de característica de dimensão 3, contendo os valores de cor no espaço CIE Lab. A distância entre os vetores de característica é calculada pela função de distância $L2$ (euclidiana).

Sendo $\mathcal{L}$ o conjunto de pixels de uma imagem, os pixels rotulados e selecionados pelo usuário formam um conjunto $\mathcal{M}$. A partir desse conjunto de treinamento são estimados os protótipos conforme descrito na Seção 2.2.1, criando os subconjuntos $\mathcal{N}_O \subset \mathcal{M}$ e $\mathcal{N}_F \subset \mathcal{M}$ de protótipos de objeto e fundo respectivamente. Cada pixel $p \in \mathcal{L}$ da imagem é então classificado como objeto ou fundo de acordo com a raiz da floresta que oferece a p o caminho ótimo no grafo (Equação 2.12). Os vetores de característica da imagem são calculados usando apenas os pixels rotulados como objeto pelo classificador OPF, ou seja, são extraídas as características da região de interesse.

Usando o conjunto $\mathcal{M}$ de pixels rotulados e os conjuntos $\mathcal{N}_O$ e $\mathcal{N}_F$, os pixels das imagens da base $\mathcal{Z}$ são classificados e as características das prováveis regiões de interesse são extraídas. Desta forma

é possível buscar um objeto específico nas imagens da base, ignorando regiões de fundo e objetos sem importância para a consulta.

É possível observar que ocorrem dois problemas principais nesta abordagem. O primeiro, é o custo computacional. Cada vez que é feita uma marcação, é necessário classificar cada um dos pixels de todas as imagens da base, além de extrair as características de cada uma delas. O outro problema é que o sistema é sensível à marcação inicial, a qual influencia diretamente a classificação do que é considerado objeto ou fundo em todas as imagens da base. Na proposta aqui apresentada, foram empregadas duas estratégias para tratar esses problemas. Para redução do tempo computacional, são utilizados *thumbnails*, ou seja, imagens reduzidas das contidas na base. Para corrigir o eventual problema na marcação inicial, foi desenvolvido um sistema de realimentação de relevância em dois níveis de interesse. A seguir são detalhadas estas duas soluções.

Para redução do tempo computacional na segmentação de objetos e na criação dos novos vetores de característica, é adotado o seguinte procedimento. Inicialmente, é criada uma versão reduzida para todas as imagens da base (*thumbnails* com 128 pixels na dimensão maior e mantendo a razão de aspecto das imagens). Cada pixel dessas versões reduzidas é classificado como objeto ou fundo, de acordo com as florestas de caminhos ótimos criadas usando o conjunto $\mathcal{M}$ de pixels rotulados e os conjuntos $\mathcal{N}_O$ e $\mathcal{N}_F$. As regiões formadas pelos pixels classificados como objeto são mapeadas na imagem de tamanho original e os vetores de característica definidos pelos pixels rotulados como objeto são calculados.

Foram realizados testes que mostraram que essa abordagem não reduz substancialmente a eficácia do método e o tempo computacional para classificar os pixels das imagens em objeto e fundo foi reduzido drasticamente pelo uso de *thumbnails*. Outra possibilidade para redução do tempo computacional seria extrair os vetores de característica diretamente dos *thumbnails*. Mas como esperado, testes mostraram que a eficácia desta segunda solução é muito inferior da obtida usando as imagens originais.

Para tratar a questão da marcação inicial, esta tese apresenta um novo método de realimentação de relevância *em dois níveis* de interesse. No nível mais geral, o usuário seleciona as imagens relevantes ou não para a sua consulta, conforme o método $GOPF_{RF}$. Além dessa seleção, em um nível de interesse mais específico, o usuário também pode ajustar as marcações de objeto e fundo em uma imagem. O sistema mostra na tela quais pixels são classificados como objeto ou fundo, ou seja, qual a provável região de interesse de acordo com a marcação original usada para extrair as características das imagens (Figura 3.5b). O usuário pode selecionar uma imagem cujo objeto ou fundo foi classificado incorretamente e fazer uma nova marcação (Figura 3.5c). Depois de corrigida a marcação, a classificação é refeita em toda a base de imagens e os novos vetores de característica são calculados, prosseguindo o processo de realimentação de relevância. Em seguida, o usuário pode então reali-

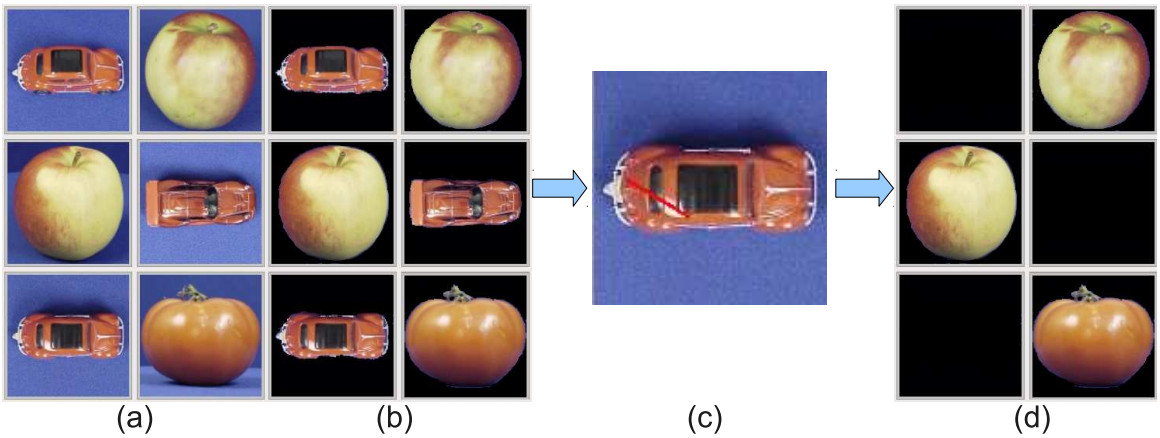


Fig. 3.5: Seleção da região de interesse: (a) imagem de exemplo, (b) somente objetos selecionados, (c) nova marcação para descartar o carro da busca e (d) objetos selecionados após a nova marcação.

zar outra marcação caso não esteja satisfeito com o resultado ou selecionar as imagens relevantes e irrelevantes, continuando o aprendizado até estar satisfeito com o resultado apresentado.

O algoritmo do método $OPF_{Bi-Level}$ assemelha-se ao Algoritmo 2, adicionando a possibilidade de refazer as marcações de pixel para selecionar os objetos de interesse. Ele utiliza o paradigma de aprendizado guloso pois é a forma apropriada de visualizar interativamente se ambos os aprendizados estão evoluindo de acordo com a necessidade do usuário. É possível verificar a cada iteração se os objetos estão sendo selecionados corretamente, possibilitando que o usuário corrija a classificação de objeto e fundo de uma imagem da base $\mathcal{Z}$, adicionando pixels rotulados no conjunto $\mathcal{M}$ e recalculando os vetores de característica de todas as imagens. Caso os objetos estejam classificados corretamente, o processo de realimentação de relevância prossegue conforme o método $GOPF_{RF}$.

Sempre que uma nova marcação de pixels é realizada, torna-se necessário refazer o cálculo dos vetores de característica de toda a base de imagens. Para todo *thumbnail* da base de imagens $\mathcal{Z}$, todo pixel $p \in \mathcal{L}$, onde $\mathcal{L}$ são as cores dos pixels de uma imagem, é classificado usando $(R, P, C, \mathcal{M}')$ e $\lambda(s)$ para $s \in \mathcal{N}_O \cup \mathcal{N}_F$ (Equação 2.12). A função lab, que converte as cores no espaço RGB para o espaço CIE Lab, é usada para extração do vetor de características na classificação dos pixels em objeto e fundo. A função $L2$ é utilizada para calcular a distância entre as cores de dois pixels. Após a classificação dos pixels em objeto ou fundo, é gerado um novo vetor de característica utilizando somente os pixels da imagem de resolução original que correspondem a uma região marcada como objeto no *thumbnail* correspondente. Para isso, pode ser utilizado qualquer descritor de imagem, como os apresentados na Seção 2.1. Os testes realizados no Capítulo 4 utilizam o descritor BIC (Stehling et al., 2002) (Seção 2.1.1).

Algoritmo 6: Algoritmo de realimentação de relevância usando OPF em dois níveis.

Entrada: Uma imagem de consulta inicial q, descritores (v_1, d_1) e (v_2, d_2) para os níveis de interesse geral e específico respectivamente, o número N de imagens retornadas por iteração e um banco de imagem $\mathcal{Z}$.

Saída: Uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens retornadas na ordem decrescente de relevância.

Auxiliares: Conjunto $\mathcal{T}$ de imagens de treinamento, mapas $R, P, C, \mathcal{T}', \mathcal{M}'$ geradas pelo classificador OPF, conjuntos $\mathcal{M}$ e $\mathcal{L}$ de pixels, conjuntos $\mathcal{S}_R \subset \mathcal{T}$, $\mathcal{S}_I \subset \mathcal{T}$, $\mathcal{N}_O \subset \mathcal{M}$ e $\mathcal{N}_F \subset \mathcal{M}$ de protótipos, conjunto $\mathcal{Y} \subset \mathcal{Z}\backslash\mathcal{T}$ de imagens classificadas como relevante pela OPF e uma lista ordenada $\mathcal{X} \subset \mathcal{Y}$ de N imagens retornadas a cada iteração.

1 $\mathcal{R} \leftarrow \emptyset$ e $\mathcal{T} \leftarrow \emptyset$.
2 Usuário seleciona uma imagem q e marca pixels de objeto e fundo, armazenado em $\mathcal{M}$.
3 Calcule os conjuntos $\mathcal{N}_O$ e $\mathcal{N}_F$ de protótipos em $\mathcal{M}$ (Seção 2.2.1) e execute $(R, P, C, \mathcal{M}') \leftarrow OPF(\mathcal{M}, \mathcal{N}_O, \mathcal{N}_F, v_2, d_2)$ (Algoritmo 1).
4 **para** *todo pixel* $p \in \mathcal{L}_q$ **faça**
5 Classifique p usando $(R, P, C, \mathcal{M}')$ e $\lambda(s)$ para $s \in \mathcal{N}_O \cup \mathcal{N}_F$ (Equação 2.12).
6 **fim**
7 Calcule o vetor de característica da região marcada como objeto na imagem q.
8 **para** *todas as imagens de* $\mathcal{Z}$ **faça** classifique os pixels dos *thumbnails* usando $(R, P, C, \mathcal{M}')$ e calcule os vetores de característica v_1 da região classificada como objeto.
9 Crie uma lista $\mathcal{X}$ com N imagens $t \in \mathcal{Z}$ mais próximas a q em ordem crescente usando d_1.
10 **enquanto** *o usuário não estiver satisfeito com* $\mathcal{X}$ **faça**
11 Usuário marca as imagens relevantes e irrelevantes, criando um conjunto de treinamento $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ onde cada imagem $s \in \mathcal{T}$ recebe rótulo de relevância $\lambda(s)$.
12 Insira as imagens relevantes de $\mathcal{X}$ no conjunto $\mathcal{R}$.
13 **se** *usuário escolhe refazer marcações de objeto/fundo* **então**
14 Usuário seleciona em uma imagem os pixels que foram erroneamente classificados como objeto ou fundo, adicionando em $\mathcal{M}$ os novos pixels rotulados.
15 Recalcule $\mathcal{N}_O$ e $\mathcal{N}_F$ e execute $(R, P, C, \mathcal{M}') \leftarrow OPF(\mathcal{M}, \mathcal{N}_O, \mathcal{N}_F, v_2, d_2)$.
16 **para** *todo pixel* $p \in \mathcal{L}_q$ **faça** classifique p usando $(R, P, C, \mathcal{M}')$.
17 Recalcule os vetores de característica de q e de todas as imagens de $\mathcal{Z}$ utilizando *thumbnails* e $(R, P, C, \mathcal{M}')$ para classificar os pixels de objeto.
18 **fim**
19 Calcule os conjuntos $\mathcal{S}_R$ e $\mathcal{S}_I$ de protótipos em $\mathcal{T}$ (usando os novos vetores de característica se for o caso) e execute $(R, P, C, \mathcal{T}') \leftarrow OPF(\mathcal{T}, \mathcal{S}_R, \mathcal{S}_I, v_1, d_1)$.
20 $\mathcal{Y} \leftarrow \emptyset$.
21 **para** *toda imagem* $t \in \mathcal{Z}\backslash\mathcal{T}$ **faça**
22 Classifique t usando $(R, P, C, \mathcal{T}')$ e $\lambda(s)$ para $s \in \mathcal{S}_R \cup \mathcal{S}_I$ (Equação 2.12).
23 **se** $\lambda(t)$ *é relevante* **então** Insira t em $\mathcal{Y}$.
24 **fim**
25 Crie uma lista $\mathcal{X}$ com as N imagens mais próximas a $t \in \mathcal{Y}$ na ordem crescente de $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.
26 **fim**
27 Retorne $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

O Algoritmo 6 apresenta o método de realimentação de relevância usando o classificador de floresta de caminhos ótimos em dois níveis utilizando o paradigma de aprendizado guloso. Ele retorna uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens na ordem decrescente de relevância. Na linha 1, a lista de treinamento $\mathcal{T}$ é inicializada com um conjunto vazio. Para uma dada imagem de consulta inicial q, o usuário seleciona inicialmente qual o objeto de interesse e remove fundo e objetos indesejados marcando pixels da imagem na linha 2. São retornados os conjuntos $\mathcal{N}_O \subset \mathcal{M}$ e $\mathcal{N}_F \subset \mathcal{M}$ de protótipos de objeto e fundo dos pixels, as listas com predecessor $P(s)$, custo $C(s)$, raiz $R(s)$ e uma lista $\mathcal{M}'$ ordenada pelo valor decrescente de custo $C(s)$. Usando estas listas, todos os pixels $p \in \mathcal{L}_q$ são classificados como objeto e fundo (linhas 4 a 6), onde $\mathcal{L}_q$ são os pixels da imagem q. Na linha 7 é recalculado o vetor de característica da imagem de consulta usando somente os pixels classificados como objeto e a função de extração de característica definida pelo descritor (v_1, d_1). Nas demais imagens, os pixels dos *thumbnails* são classificados usando $(R, P, C, \mathcal{M}')$ e os pixels mapeados na imagem de resolução normal considerados como objeto pela OPF são usados para gerar seus novos vetores de característica (linha 8). Usando a imagem q segmentada e a base de imagens $\mathcal{Z}$, o algoritmo retorna inicialmente uma lista $\mathcal{X}$ com as N imagens $t \in \mathcal{Z}$ mais próximas a q na ordem crescente de acordo com a função de distância $d_1(q, t)$ (linha 9). O processo de aprendizado é executado no laço principal (linhas 10 a 26).

O usuário marca, na linha 11, as imagens de $\mathcal{X}$ que ele considera relevante ou irrelevante, prosseguindo com o processo de realimentação de relevância conforme já apresentado no Algoritmo 2, inserindo as imagens relevantes no conjunto $\mathcal{R}$ (linha 12).

Se o usuário verificar que a marcação necessita sofrer algum ajuste como exemplificado na Figura 3.5, ele seleciona na linha 14 uma das imagens cujo objeto ou fundo esteja marcado incorretamente e refaz a seleção. Os pixels marcados são adicionados ao conjunto $\mathcal{M}$ e o classificador OPF é executado na linha 15, onde v_2 contém o valor CIE Lab do pixel e d_2 é calculado pela função $L2$. Os mapas $(R, P, C, \mathcal{M}')$ criados na linha 15 são usados para classificar os pixels da imagem q (linha 16). Depois são recalculados os vetores de característica da imagem q e de todas as imagens da base $\mathcal{Z}$, classificando os pixels de seus *thumbnails*. Usando os pixels da imagem de resolução normal cuja posição mapeada no seu *thumbnail* foi classificada como objeto, são extraídas as suas características (linha 17).

Caso não seja necessário fazer o ajuste das regiões de objeto e fundo, o algoritmo utiliza os vetores de característica calculados anteriormente. Senão, os novos vetores de característica são utilizados no processo de realimentação de relevância (linha 19), que prossegue conforme o Algoritmo 2. Ao final do processo (linha 27), são retornadas as imagens rotuladas como relevantes pelo usuário durante a realimentação de relevância (conjunto $\mathcal{R}$) mais N imagens candidatas a relevantes.

Os resultados dos experimentos realizados utilizando $OPF_{Bi-Level}$ são apresentados na Seção 4.5, mostrando exemplos de execução deste novo método de realimentação de relevância.

Capítulo 4

Resultados

Do ponto de vista prático, um sistema de recuperação de imagens por conteúdo baseado no aprendizado por realimentação de relevância deve necessitar de poucas iterações para aprendizagem (*eficácia*) e retornar as imagens em tempo interativo (*eficiência*). Estes constituem os principais desafios na recuperação de informação, especialmente quando consideramos grandes coleções de imagens. Este capítulo demonstra que os métodos aqui apresentados são eficazes e eficientes se comparados aos métodos considerados como sendo o estado-da-arte em realimentação de relevância.

Para verificar a eficácia de um sistema de recuperação de imagens, é necessário usar uma métrica para comparar dois ou mais métodos. A medida mais comumente usada para avaliar a eficácia de um sistema de recuperação de informação é a curva precisão vs. revocação ($P \times R$). Para uma imagem de consulta inicial q, a precisão $P(q)$ representa o número de imagens relevantes retornadas $Rel(q)$ em relação ao número total de imagens retornadas $N(q)$ (Equação 4.1). Se o valor de precisão obtida para uma certa consulta for de 70%, sabe-se que 70% das imagens retornadas são relevantes e as demais 30% são irrelevantes. Revocação $R(q)$ é o número de imagens relevantes retornadas em relação ao número total de imagens relevantes na base $M(q)$ para uma dada consulta q (Equação 4.2). Se o valor de revocação obtido para uma certa consulta for de 50%, sabe-se que 50% das imagens relevantes de toda base foram retornadas para o conjunto de resultados em questão.

$$P(q) = \frac{Rel(q)}{N(q)} \tag{4.1}$$

$$R(q) = \frac{Rel(q)}{M(q)} \tag{4.2}$$

Em uma curva $P \times R$, quanto mais alta uma curva, mais eficaz é a recuperação de informação, pois a curva $P \times R$ indica a variação dos valores de precisão para diferentes valores de revocação

(10%, 20%,...). Como é necessário conhecer previamente a quantidade de imagens relevantes para cada tipo (classe) de imagem, essa curva precisa ser calculada em uma base de imagens previamente conhecida e classificada. Esta abordagem permite comparar os resultados dos diferentes algoritmos nas mesmas condições de teste.

O outro critério de avaliação utilizado é a curva percentual de imagens relevantes retornadas por iteração ($Rel \times It$). Nesta curva é apresentado o percentual de imagens relevantes exibidas ao usuário em cada iteração realizada. Esta curva permite avaliar como o número de imagens relevantes retornadas aumenta ao longo das iterações. Já que todas as imagens da base são utilizadas para a geração das curvas, a linha 14 dos Algoritmos 2 e 3 (Capítulo 3) foi alterada para: *Retorne* $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{Y}$ *ordenada por* $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ para geração das curvas $Rel \times It$.

Para avaliar a eficácia tanto dos métodos desenvolvidos na tese quanto das técnicas tradicionais, essas curvas são calculadas sobre bases de imagens consolidadas comumente usadas na área de CBIR. Optou-se pelo uso de bases de imagens rotuladas simulando a experiência de usuários na busca de imagens. Outra possibilidade seria quantificar os resultados obtidos por usuários em experimentos de campo, mas esta estratégia é muito custosa operacionalmente pois deveria envolver um número grande de usuários para ser estatisticamente confiável, já que é necessário levar em consideração alguns fenômenos psicológicos que influenciam na seleção das imagens (Joachims e Radlinski, 2007), como a ordem das imagens que são apresentadas a cada iteração, o efeito de pareidolia, tempo que o usuário leva para a escolha das imagens, entre outros. Além disso, usuários diferentes tendem a ter respostas diferentes à mesma pergunta e não é raro que o mesmo usuário tenha respostas variadas em diferentes momentos. A seguir são apresentadas as bases de imagens usadas nos experimentos desta tese.

Neste capítulo são exibidas as curvas médias calculadas considerando todas as imagens de uma base de imagens através da técnica de validação cruzada *leave-one-out*[1]. Depois de calculada a curva para todas as imagens da base, é calculada a média do resultado de precisão $P(q)$ que define a curva apresentada na avaliação comparativa entre os diferentes métodos de realimentação de relevância.

4.1 Bases de imagens

A fim de avaliar a eficácia dos métodos desenvolvidos nesta tese, buscou-se utilizar bases de imagens de propósito geral e que sirvam como diferentes desafios para a área de recuperação de imagens por conteúdo. Neste sentido, foram utilizadas as seguintes bases de imagens:

[1]Cada imagem da base é usada como a imagem de consulta inicial q e as demais imagens da base são usadas como espaço de consulta

- Caltech 101 (Fei-Fei et al., 2004)

  Imagens de 9.144 objetos pertencentes a 101 categorias, com 40 a 800 imagens por categoria e a maioria das categorias com aproximadamente 50 imagens.

- Coil-100 (Nene et al.)

  É uma base de imagens composta de 100 objetos. Foi formada a partir de fotografias tiradas de cada um dos objetos em 72 poses diferentes, num total de 7.200 imagens.

- Corel (Corel Corp.)

  É uma coleção de 200.000 imagens da *GALLERY Magic–Stock Photo Library 2*. É utilizada nesta tese um subconjunto composto por 3.906 exemplos, classificados em 85 classes. Essas classes são de diferentes tamanhos, variando de 7 a 98 imagens em cada classe.

- ETH-80 (Leibe e Schiele, 2003).

  Este banco de imagens foi desenvolvido para o projeto *COGVIS*[2]. O projeto inclui imagens de objetos de oito categorias (automóvel, cachorro, cavalo, maçã, pera, tomate, vaca e xícara), num total de 2.384 imagens distribuídas uniformemente entre as classes.

- MPEG7 (MPEG7 CE Shape-1 Part B) (Sikora, 2001)

  É uma base de 1.400 imagens binárias de forma com 70 categorias, sendo 20 objetos por categoria.

- MSRCORID (Microsoft Research Cambridge)

  Contém um conjunto de 4.320 imagens agrupadas em 20 categorias, sendo 36 a 652 imagens por categoria e a maioria das classes com aproximadamente 200 imagens.

- PASCAL (Everingham et al.)

  Essa base é formada por imagens do Flickr [3] e é usada no desafio *PASCAL Visual Object Classes (VOC)*. Nesta tese foi utilizado um subconjunto de 3.448 objetos agrupados em 23 categorias, possuindo de 72 a 446 subimagens cada.

[2]http://www.cogvis.at/
[3]www.flickr.com

4.2 Exemplo de execução da técnica de realimentação de relevância

Para ilustrar a utilização da técnica de realimentação de relevância, discutida nesta tese, é apresentado um exemplo de execução usando o paradigma guloso. Neste exemplo, são apresentadas $N = 30$ imagens a cada iteração usando a base de imagens Corel e o descritor BIC (Stehling et al., 2002) (Seção 2.1.1).

O teste da eficácia dos métodos de realimentação de relevância desenvolvidos nesta tese usando um único descritor ($GOPF_{RF}$ e $POPF_{RF}$ na Seção 4.3)[4] é feito comparando os resultados obtidos por esses métodos com os resultados de dois outros métodos comumente usados para realimentação de relevância: QEX (*Query Expansion method*) (Porkaew et al., 1999)[5] e SVM_{AL} (*SVM Active Learning*) proposto por Tong e Chang (2001)[6].

QEX designado como QPM por Tong e Chang (2001) utiliza a abordagem gulosa usando apenas imagens relevantes no aprendizado, diferentemente dos métodos desenvolvidos nesta tese que também utilizam as imagens irrelevantes. Neste método, as imagens marcadas pelo usuário como relevantes em $\mathcal{X}$ formam um conjunto $\mathcal{R} \subset \mathcal{X}$. As imagens em $\mathcal{R}$ formam agrupamentos e os centroides q_i desses agrupamentos são utilizados para obter as imagens das próximas iterações. As N imagens $t \in \mathcal{Z} \backslash \mathcal{R}$ ordenadas pela distância $d(t, q_i)$ entre t e cada um dos centroides são escolhidas para formar um novo conjunto $\mathcal{X}$ (Figura 2.4b à esquerda). As imagens são, portanto, as mais próximas a qualquer um dos centroides q_i. O resultado depois de I iterações é apresentado em $\mathcal{R} \cup \mathcal{X}$, sendo $\mathcal{X}$ o último conjunto de N imagens retornadas como candidatas a relevante.

SVM_{AL} (também conhecido por SAL ou SVM_{ACTIVE}) utiliza abordagem semelhante à planejada definida neste trabalho através de um classificador SVM. Apesar de existirem outros modelos baseados nesta técnica, este ainda é o mais utilizado para comparação de resultados e considerado estado-da-arte em realimentação de relevância. Conforme descrito no capítulo anterior, o laço externo do paradigma planejado (Algoritmo 3) é executado uma única vez nos métodos baseados em SVM. Desta forma, os resultados para o método $POPF_{RF}$ são obtidos também executando este laço uma única vez simulando o comportamento do usuário. No método SVM_{AL}, as imagens relevantes e irrelevantes em $\mathcal{X}$ formam, a cada iteração, o conjunto de treinamento $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$. O processo de treinamento consiste em encontrar vetores de suporte em $\mathcal{T}$ calculando o hiperplano que melhor separa os conjuntos relevantes e irrelevantes em $\mathcal{T}$ neste espaço de características. Durante as I iterações do laço interno, SVM_{AL} retorna em $\mathcal{X}$ as N imagens $t \in \mathcal{Z} \backslash \mathcal{T}$ mais próximas ao hiperplano, que representam as imagens mais informativas. As imagens mais próximas ao hiperplano são aquelas

[4]Métodos implementados usando a LibOPF-2.0 disponível em http://www.ic.unicamp.br/ afalcao/libopf/

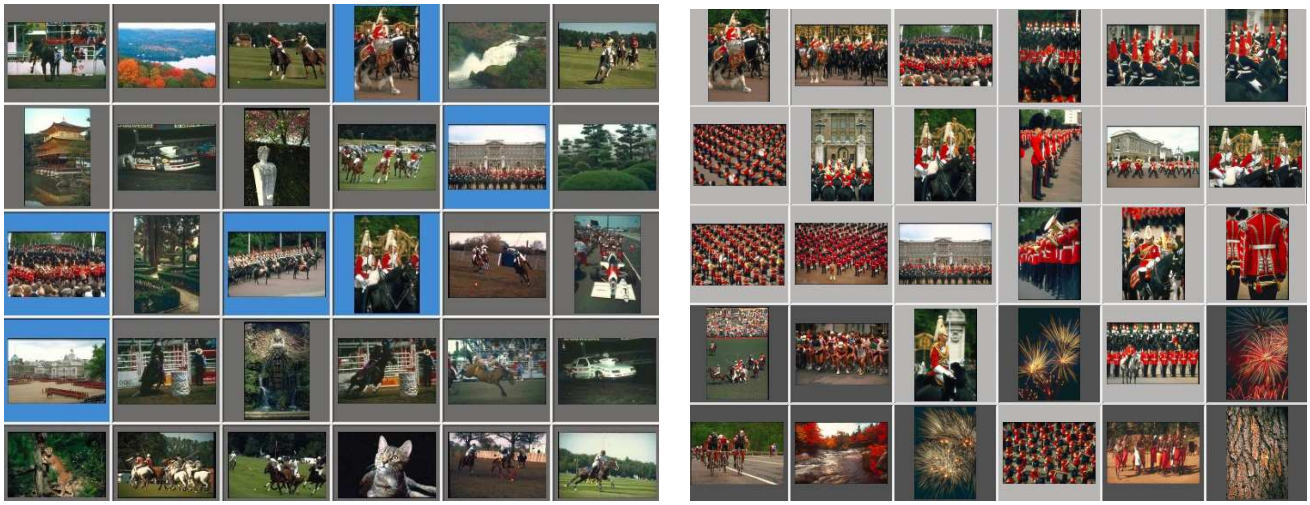
[5]Método implementado conforme proposto pelo autor

[6]Método implementado usando a libsvm-2.83 disponível em http://www.csie.ntu.edu.tw/ cjlin/libsvm/

mais difíceis de se classificar, ou seja, as que estão muito próximas de pertencer tanto à classe de imagens relevantes quanto à classe de irrelevantes. As imagens mais afastadas tem maior possibilidade de pertencer à classe de uma região e menor probabilidade de pertencer à classe da região oposta. Ao final do processo, após a I-ésima iteração, são retornadas em $\mathcal{X}$ as N imagens mais afastadas do hiperplano no lado das relevantes.

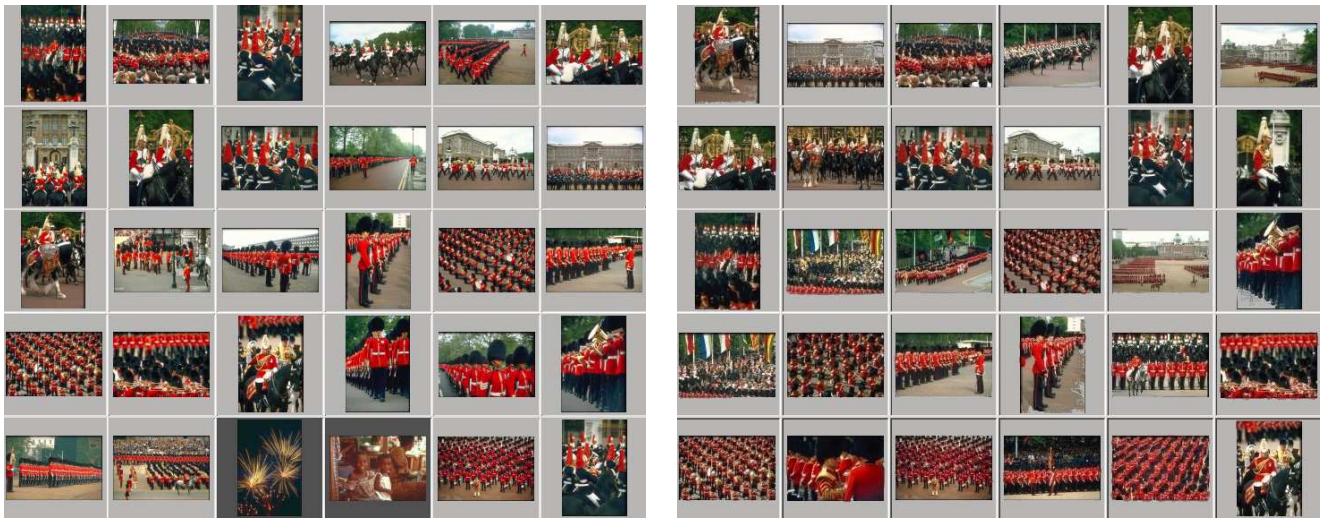
Neste exemplo ilustrativo, a partir da imagem da Figura 4.1 informada pelo usuário, inicialmente são retornadas as imagens da Figura 4.2a em ordem crescente de (v, d). Em todas as figuras a relevância definida por (v, d) decresce da esquerda para a direita e então de cima para baixo. As imagens marcadas em azul representam as imagens consideradas pelo usuário como relevantes. Após 3 iterações (um número razoável para uma situação prática) são apresentadas as 30 primeiras imagens (mais relevantes) para QEX (Figura 4.2b), SVM_{AL} (Figura 4.3a) e $POPF_{RF}$ (Figura 4.3b). Considerando a classificação definida para a base Corel, em QEX apenas 21 das images são realmente relevantes enquanto SVM_{AL} retornou 28 e em $POPF_{RF}$ todas as 30 imagens são relevantes.

Fig. 4.1: Imagem de consulta inicial.



(a) (b)

Fig. 4.2: (a) Imagens apresentadas após a primeira iteração e (b) 30 primeiras imagens apresentadas após a terceira iteração para QEX.



(a) (b)

Fig. 4.3: (a) 30 primeiras imagens apresentadas após a terceira iteração para SVM_{AL} e (b) 30 primeiras imagens apresentadas após a terceira iteração para $GOPF_{RF}$.

4.3 Resultados de $GOPF_{RF}$ e $POPF_{RF}$

Esta Seção compara experimentos realizados com os métodos $GOPF_{RF}$ (Seção 3.1) e $POPF_{RF}$ (Seção 3.1) utilizando os métodos QEX e SVM_{AL} como referência.

A eficácia dos métodos $GOPF_{RF}$ e $POPF_{RF}$ é comparada através do resultado das curvas $P \times R$. Os gráficos $P \times R$ são gerados através da técnica de validação cruzada *leave-one-out* simulando o comportamento do usuário para todas as imagens $q \in \mathcal{Z}$ e em cada iteração (e interação simulada) são apresentadas $N = 30$ imagens. Em uma execução real de um sistema de realimentação de relevância, o laço inicial termina quando o usuário está satisfeito com a busca realizada. Para gerar os resultados aqui apresentados, no entanto, é fixado um número I de iterações. Da mesma forma, para os métodos que usam o paradigma planejado, o laço interno executa I iterações enquanto o laço externo é realizada apenas uma vez para realizar a comparação com o método SVM_{AL}.

As Figuras 4.4, 4.5 e 4.6 apresentam a curva $P \times R$ média para cada um dos métodos ($POPF_{RF}$, $GOPF_{RF}$, SVM_{AL} e QEX) para $I = 3, 5, 8$ iterações usando a base Corel. É possível observar que ambos os métodos de realimentação de relevância baseados em floresta de caminhos ótimos superam a eficácia dos outros métodos em qualquer um dos diferentes números de iteração. Nota-se também que o método usando a abordagem planejada aprende mais rápido, ou seja, necessita de menos iterações para recuperar a mesma quantidade de imagens relevantes. Por exemplo, os métodos propostos nesta tese necessitam de apenas 3 iterações para obter uma média de 90% de precisão para 30% de revocação (30% das imagens relevantes da base). É muito provável que o usuário esteja satisfeito neste momento em uma situação real. Os outros métodos apresentaram em média uma precisão pelo menos 10% inferior para os mesmos 30% de revocação após a terceira iteração, conforme é mostrado na Figura 4.4.

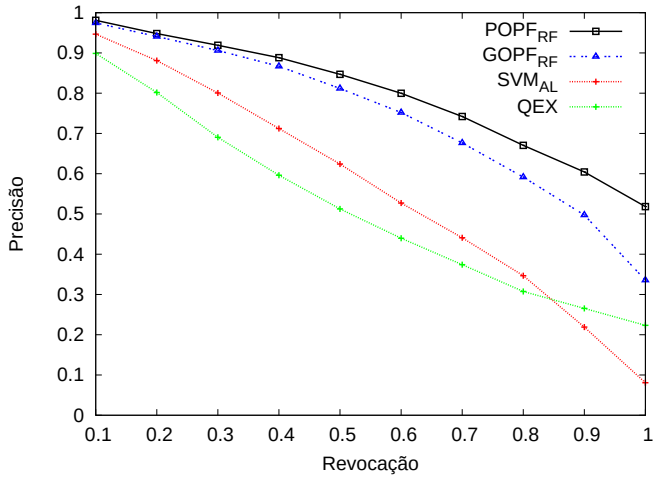


Fig. 4.4: Curva $P \times R$ média na base Corel após a terceira iteração.

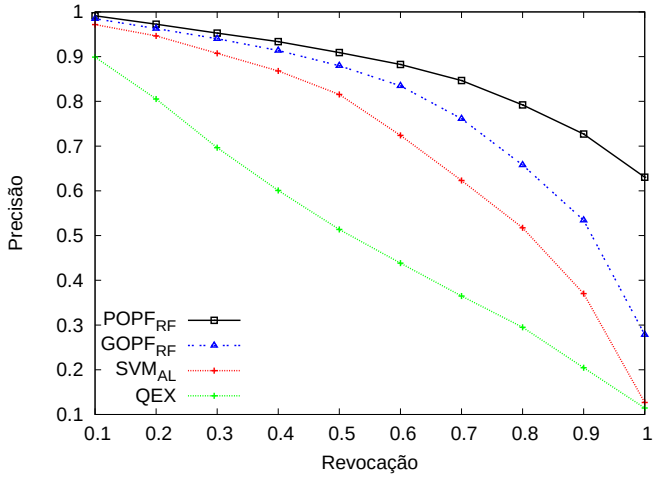


Fig. 4.5: Curva $P \times R$ média na base Corel após a quinta iteração.

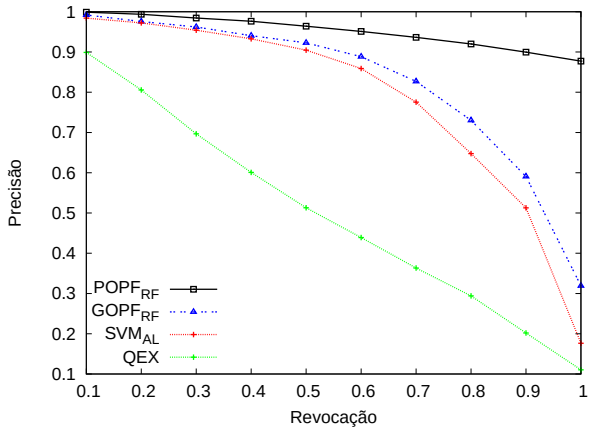


Fig. 4.6: Curva $P \times R$ média na base Corel após a oitava iteração.

O ganho em eficácia dos métodos desenvolvidos nesta tese também pode ser visto para as demais bases de imagens. As Figuras 4.7 até 4.10 mostram a curva média $P \times R$ para cada um dos métodos utilizando as bases de imagens Caltech, Coil-100, MSRCORID e Pascal, respectivamente, para 3 e 8 iterações de realimentação de relevância. Considerando que 3 iterações é um bom número para situações reais, nota-se que os métodos propostos apresentam um ganho considerável neste caso. A base Caltech é a mais difícil para o descritor BIC, enquanto a Coil-100 é a mais fácil. QEX superou SVM_{AL} na base Coil-100 e o mesmo ocorreu na base Pascal até a terceira iteração. De fato, SVM_{AL} necessita mais iterações para melhorar a eficácia a fim de obter um bom conjunto de treinamento. Por este motivo, alguns trabalhos (Hoi e Lyu, 2005; Hoi et al., 2010) incluem imagens não rotuladas pelo usuário no conjunto de treinamento.

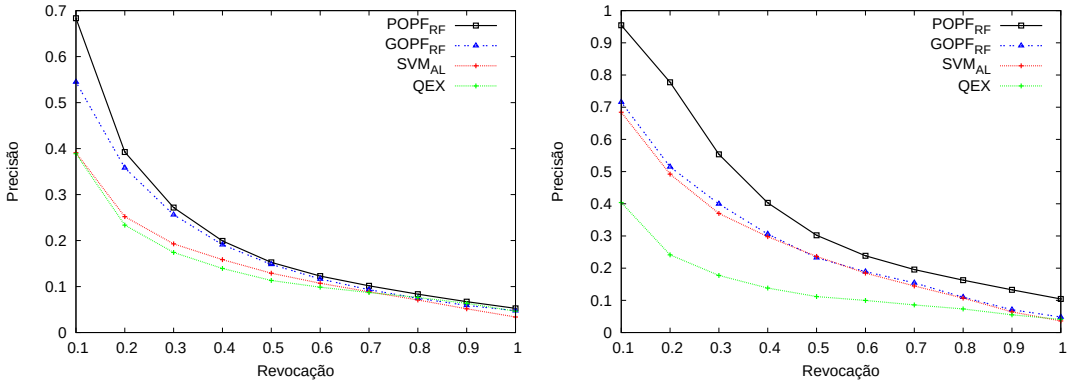


Fig. 4.7: Curva $P \times R$ média na base Caltech após (a) terceira e (b) oitava iterações.

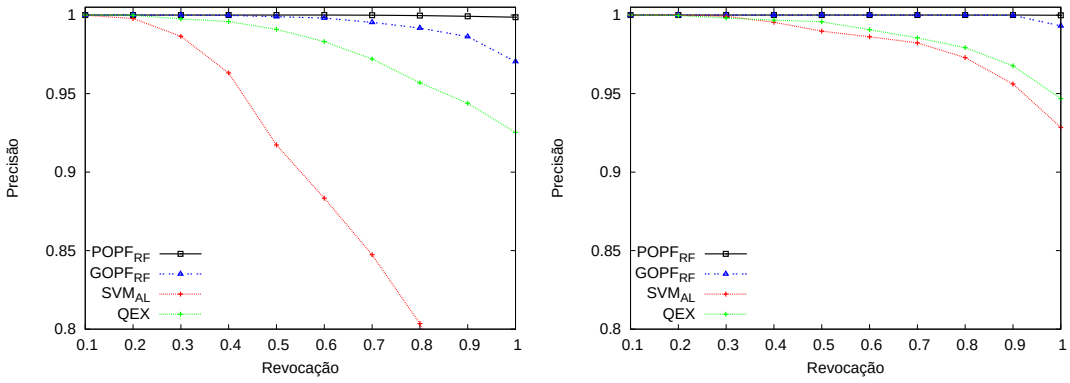


Fig. 4.8: Curva $P \times R$ média na base Coil-100 após (a) terceira e (b) oitava iterações.

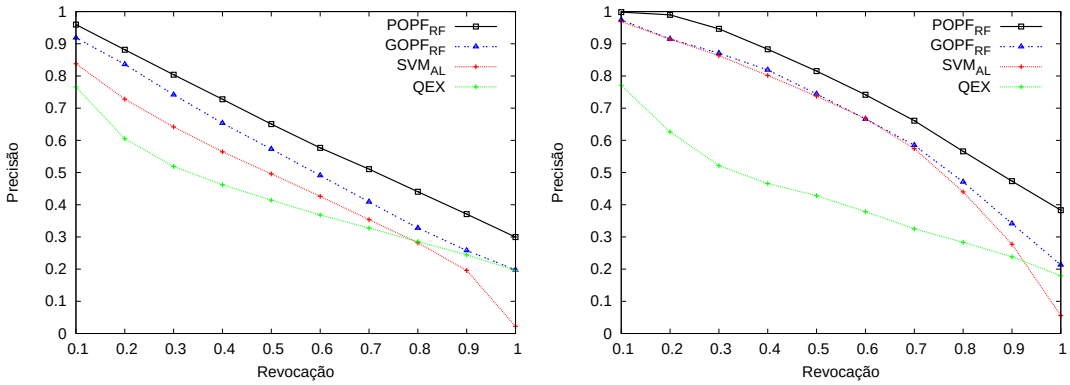


Fig. 4.9: Curva $P \times R$ média na base MSRCORID após (a) terceira e (b) oitava iterações.

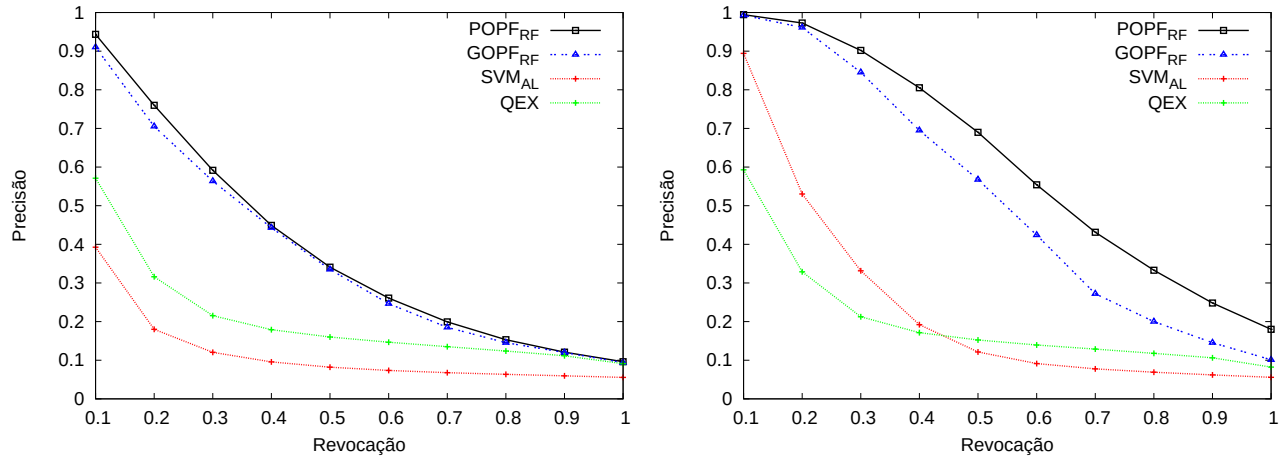


Fig. 4.10: Curva $P \times R$ média na base Pascal após (a) terceira e (b) oitava iterações.

A eficiência de todos os métodos foi avaliada através da execução dos experimentos que geraram as curvas apresentadas nas Figuras 4.4 a 4.10. A Tabela 4.1 mostra o tempo total de execução para oito iterações em cada uma das bases. O tempo de execução médio para cada iteração é apresentado na Tabela 4.2. Os resultados mostram que o método SVM_{AL} é o mais caro computacionalmente, devido ao treinamento da SVM. É importante observar que não foram utilizadas estruturas de indexação para acelerar a busca, necessárias para bases de imagens muito grandes. Pode-se afirmar que considerando as bases de imagens utilizadas os tempos de execução para os métodos $GOPF_{RF}$ e $POPF_{RF}$ são em média 52 vezes menores que os tempos do método SVM_{AL}. Na base Corel, por exemplo, tanto $GOPF_{RF}$ quanto $POPF_{RF}$ levam cerca de 0,1 segundos para apresentar as imagens após a oitava iteração, enquanto SVM_{AL} leva em média 8,9 segundos. Isto confirma o ganho em eficiência do classificador OPF sobre SVM, conforme reportado por Papa et al. (2009). Também são apresentados os tempos de execução para a base PASCAL utilizando todas as 31.284 imagens (PASCAL*) para mostrar o desempenho dos métodos em uma base de imagens maior. Os resultados de tempo de execução para o método SVM_{AL} não são mostrados para a base PASCAL* pelo fato deste método ser muito lento.

O método QEX tem uma eficiência similar aos métodos baseados em floresta de caminhos ótimos. É possível observar que ele obtém um desempenho melhor nas bases de imagens mais difíceis para CBIR. Como ele utiliza apenas as imagens marcadas como relevantes durante as iterações, quanto menor o número de imagens relevantes, maior o desempenho do método já que são menos pontos de consulta para o aprendizado. É importante observar que justamente nas bases cuja eficiência de QEX é melhor, sua eficácia é inferior à dos outros métodos (Figuras 4.7, 4.9 e 4.10).

Os testes foram realizados em um computador com processador Intel Core i7 a 2,8 GHz e 8 GB

de memória RAM, executando o sistema operacional Linux.

Tab. 4.1: Tempo total de execução para 8 iterações e todas as imagens de consulta (minutos).

Base	Caltech	Coil-100	Corel	MSRCORID	PASCAL	PASCAL*
QEX	169,5	184,3	54,7	60,5	25,7	2.871,5
SVM_{AL}	7.229,8	4.608,0	3.312,3	3.749,8	2.744,6	—
$GOPF_{RF}$	178,0	101,7	37,4	58,1	35,8	3.936,4
$POPF_{RF}$	178,0	101,8	37,5	58,2	35,9	3.958,3

Tab. 4.2: Tempo médio de execução por imagem de consulta (segundos).

Base	Caltech	Coil-100	Corel	MSRCORID	PASCAL	PASCAL*
QEX	0,14	0,20	0,11	0,06	0,05	0,69
SVM_{AL}	5,93	4,80	6,36	6,51	5,97	—
$GOPF_{RF}$	0,15	0,11	0,07	0,10	0,08	0,94
$POPF_{RF}$	0,15	0,11	0,07	0,10	0,08	0,95

4.4 Resultados de OPF_{MSPS} e OPF_{GP}

Esta Seção mostra os resultados obtidos pela utilização dos métodos OPF_{MSPS}, OPF_{GP} (Seções 3.3 e 3.4, respectivamente) que usam o conceito de descritor composto juntamente com o método baseado em OPF.

Os resultados são comparados com os de GP^+ (Ferreira et al., 2011), um método baseado em programação genética que pode ser considerado como estado-da-arte para combinação de descritores. Também é mostrada a curva $P \times R$ usando apenas o melhor dos descritores. A comparação com GP^+ avalia a eficiência dos métodos na combinação de descritores, enquanto a comparação aos resultados do melhor descritor e GP^+ mostra a importância do classificador OPF para a recuperação de imagens. Além disso, a combinação de descritores é justificada apenas se sua eficácia for superior à utilizar apenas um descritor. Esta é a razão de apresentar a comparação dos resultados de OPF_{MSPS} e OPF_{GP} aos obtidos usando um único descritor.

GP^+ visa buscar imagens mais relevantes aplicando um esquema de votação através dos melhores indivíduos. Neste esquema de votação, os melhores indivíduos da última geração votam para N imagens candidatas. As imagens mais votadas são as escolhidas para formar o conjunto $\mathcal{X}$ apresentado ao usuário a cada iteração do processo de realimentação de relevância. GP^+ supera outros métodos de realimentação de relevância (Rui et al., 1998; Rui e Huang, 2000; Tong e Chang, 2001; Doulamis e

Doulamis, 2006; Min e Cheng, 2009). Por este motivo, este método foi escolhido para ser comparado com os métodos desenvolvidos nesta tese.

Para integrar GP^+ aos métodos propostos nesta tese seria necessário adaptar seu esquema de votação que escolhe as imagens da próxima iteração dentre as mais votadas, o que conflitaria com a métrica aqui proposta que escolhe as imagens usando a Equação 3.2. Não foi possível em um primeiro momento combinar GP^+ ao método baseado em OPF e por isso foi utilizado o método GP descrito na Seção 3.4 nesta combinação.

As curvas $P \times R$ são mostradas para 3, 5 e 8 iterações usando as seis bases de imagens heterogêneas que representam diferentes desafios para CBIR: Coil-100, Corel, ETH-80, MPEG7, MSRCORID e PASCAL. Também são apresentadas as curvas de percentual de imagens relevantes retornadas por iteração para OPF_{MSPS}, OPF_{GP} e GP^+ para as mesmas bases de imagens. Ambas as curvas foram calculadas considerando $N = 30$ imagens por iteração.

Foram utilizados diversos descritores para os testes de combinação. Os resultados apresentados neste capítulo utilizam os descritores apresentados na Seção 2.1. De todos os descritores testados, foram escolhidos os melhores para combinação conforme mostrado na Tabela 4.3. Para as bases Corel, MSRCORID e Pascal foram selecionados três descritores de cor e dois para informação de textura. Para ETH-80, que inclui informação de forma, foram utilizados dois descritores de cor, um de textura e outros dois de forma. Como a base MPEG7 só tem informação de forma, foram utilizados três descritores para esta característica. Os métodos descritos nesta tese apresentam resultados muito eficientes para a base de imagens Coil-100 usando um bom descritor, como pode ser visto na Figura 4.8. Por isso, foram utilizados os três descritores menos eficazes para testar a eficácia dos métodos para combinação de descritores (SID, LBP e Color Bitmap) para esta base.

Tab. 4.3: Descritores combinados em cada base de imagens.

Base	Descritor
Coil-100	SID, LBP e Color Bitmap
Corel	ACC, BIC, JAC, LAS e SASI
ETH-80	ACC, BIC, LAS, MSF e TSDIZ
MPEG7	Fourier, MSF e TSDIZ
MSRCORID	ACC, BIC, JAC, LAS e SASI
PASCAL	ACC, BIC, JAC, LAS e SASI

Para o método OPF_{MSPS} foram utilizados os mesmos cinco valores de escala (0,001; 0,01; 0,12; 0,45; e 1,0) para os deslocamentos $\Delta_{i,j}$ em todos os parâmetros $i = 1, 2, \ldots, n$ para todas as bases de imagens testadas. Foram testadas mais escalas, mas isto não refletiu em aumento de eficiência ou eficácia. Com mais escalas, o método chega a uma boa combinação em menos iterações no

Algoritmo 4, mas são realizados mais testes em cada uma delas. O método baseado em programação genética é sensível aos parâmetros iniciais (ver Tabelas 4.4 e 4.5). Por isso, foram utilizados diferentes valores de tamanho de população e número de gerações para cada uma das bases testadas tanto para OPF_{GP} quanto para GP^{+}. A Tabela 4.4 mostra os valores de parâmetros (Koza, 1992) usados pelo algoritmo de programação genética em OPF_{GP}. A Tabela 4.5 apresenta o tamanho da população e o número de gerações do algoritmo de programação genética de OPF_{GP} para cada uma das bases de imagens. Os valores de parâmetros utilizados para o GP^{+} são os mesmos usados para a base Corel em Ferreira et al. Ferreira et al. (2011).

Tab. 4.4: Valores dos parâmetros para GP do método OPF_{GP}.

população inicial	*half-and-half*
profundidade inicial da árvore	$2-5$
profundidade máxima da árvore	5
método de seleção	tournament (size 2)
probabilidade de *crossover*	$0,8$
probabilidade de mutação	$0,9$
probabilidade de reprodução	$0,05$
conjunto de funções	$+, *, \sqrt{}$

Tab. 4.5: Tamanho da população e número de gerações para GP do método OPF_{GP} para cada base de imagens.

	n_p	n_g
Coil100	40	8
Corel3906	100	10
Eth80	100	10
Mpeg7	100	10
msrcorid	50	10
Pascal	60	10

As Figuras 4.11 a 4.19 mostram através da curva média de $P \times R$ para 3, 5 e 8 iterações de realimentação de relevância usando $N = 30$ imagens por iteração a evolução do aprendizado de cada método em cada base de imagens. É possível observar que ambos OPF_{MSPS} e OPF_{GP} superam GP^{+} em todas as bases testadas. OPF_{MSPS} supera OPF_{GP} nas bases Coil-100 e Pascal enquanto OPF_{GP} é mais efetivo nas bases ETH-80 e MSRCORID. Na base Corel, OPF_{MSPS} teve um melhor

resultado até a quinta iteração e OPF_{GP} superou OPF_{MSPS} após oito iterações. Na base de formas MPEG7, OPF_{MSPS} teve um bom resultado para três iterações enquanto OPF_{GP} superou OPF_{MSPS} a partir da quinta iteração.

Além disso, pode-se observar que OPF_{RF} usando apenas um descritor (TSDIZ para a base de forma MPEG7, Color Bitmap para a base Coil-100 e BIC para as demais) foi mais eficiente do que GP^{+} em vários resultados, mostrando que a fase de classificação é muito importante no processo de realimentação de relevância.

As Figuras 4.20 a 4.22 apresentam a curva média do percentual de imagens relevantes retornadas por iteração para as mesmas bases de imagens mostrando a evolução do aprendizado da primeira à oitava iteração. Estas curvas confirmam que OPF_{MSPS} e OPF_{GP} superam GP^{+} em todas as bases ao longo de todas as iterações testadas. Pode-se observar que na base Corel, OPF_{GP} passa a superar OPF_{MSPS} entre a sexta e a sétima iteração, enquanto na base MPEG7 OPF_{GP} passa a superar OPF_{MSPS} entre a terceira e a quarta iteração.

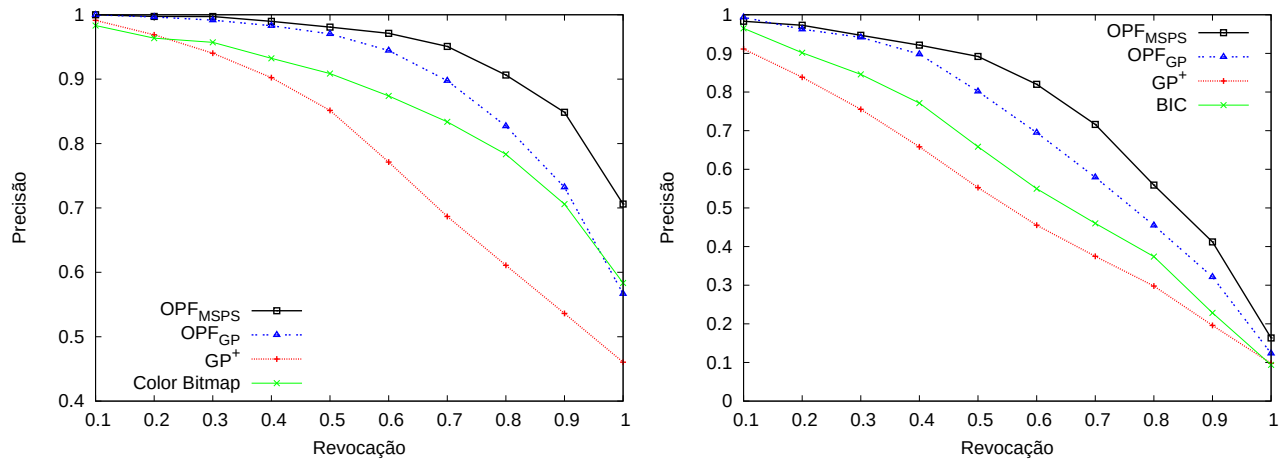


Fig. 4.11: Curva $P \times R$ média nas bases (a) Coil-100 e (b) Corel após a terceira iteração.

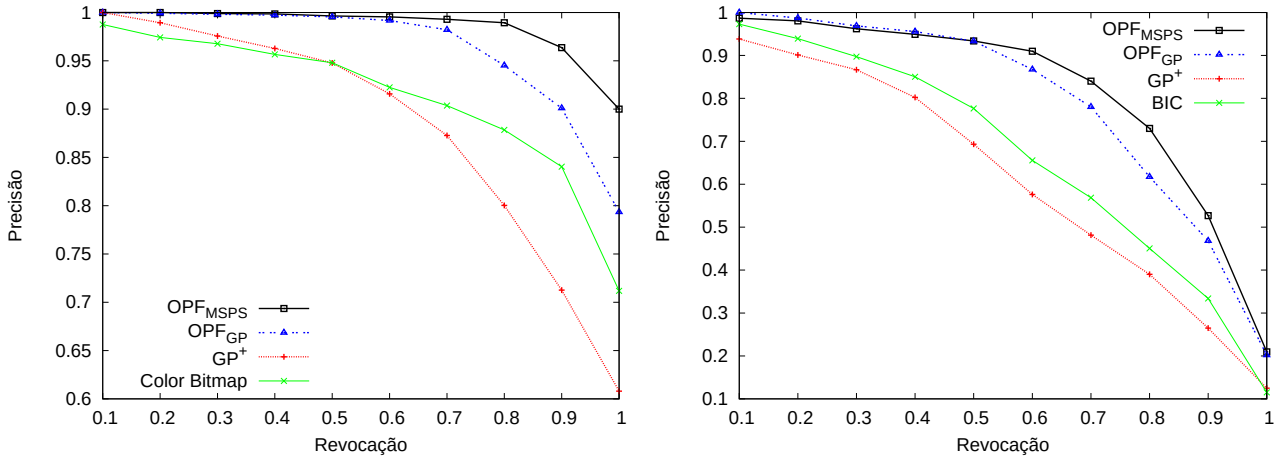


Fig. 4.12: Curva $P \times R$ média nas bases (a) Coil-100 e (b) Corel após a quinta iteração.

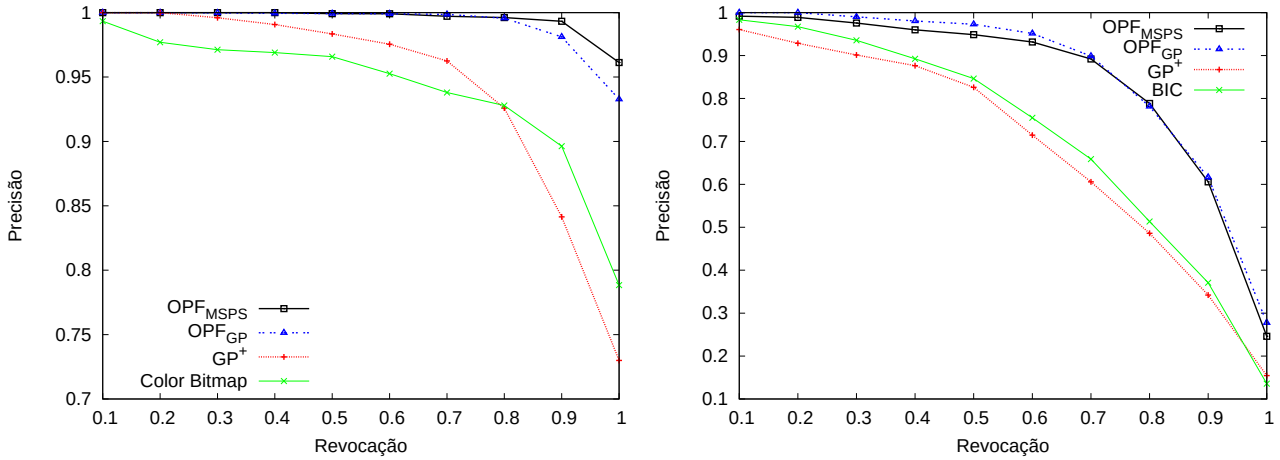
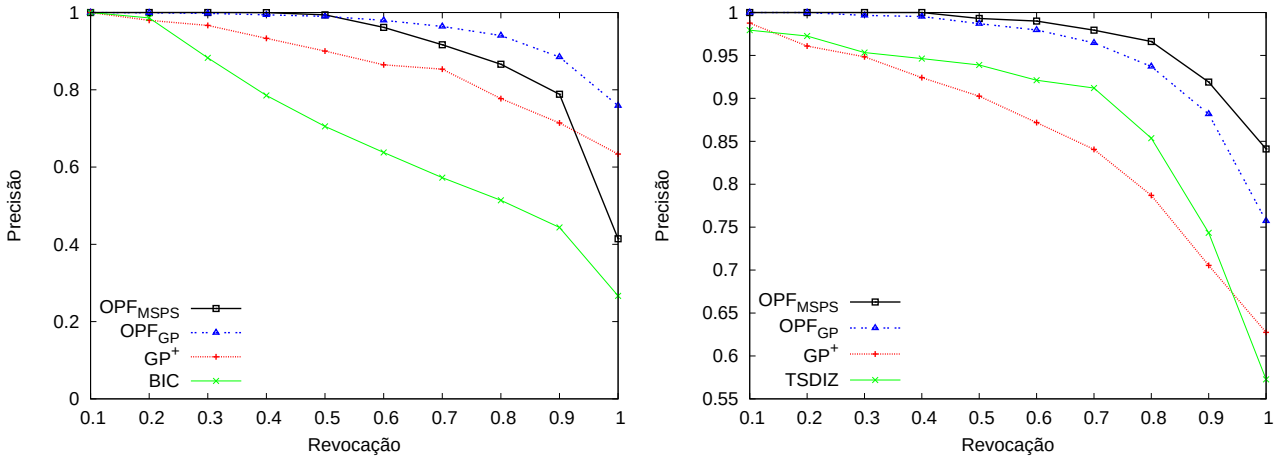
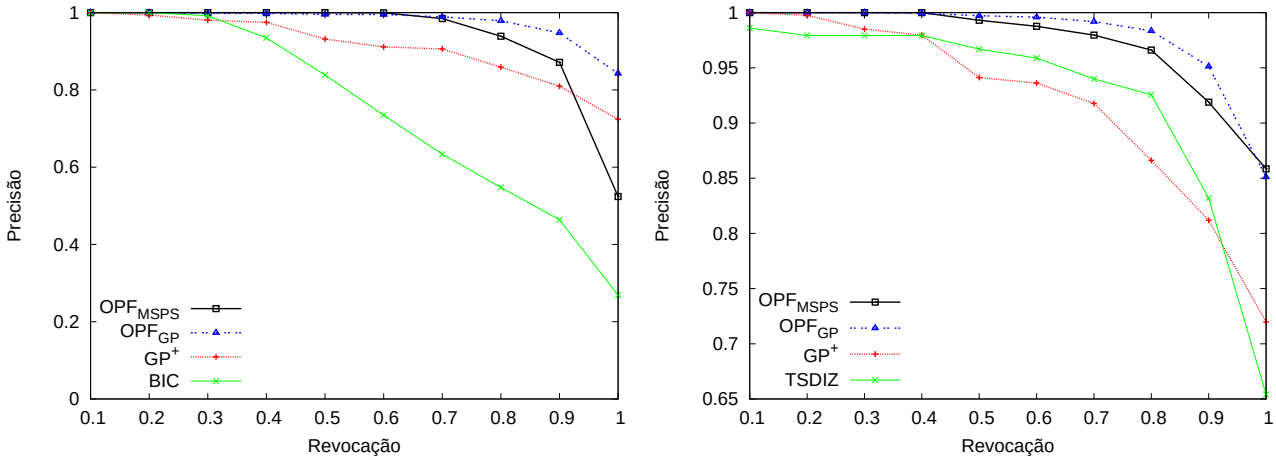


Fig. 4.13: Curva $P \times R$ média nas bases (a) Coil-100 e (b) Corel após a oitava iteração.



(a) (b)

Fig. 4.14: Curva $P \times R$ média nas bases (a) ETH-80 e (b) MPEG7 após a terceira iteração.



(a) (b)

Fig. 4.15: Curva $P \times R$ média nas bases (a) ETH-80 e (b) MPEG7 após a quinta iteração.

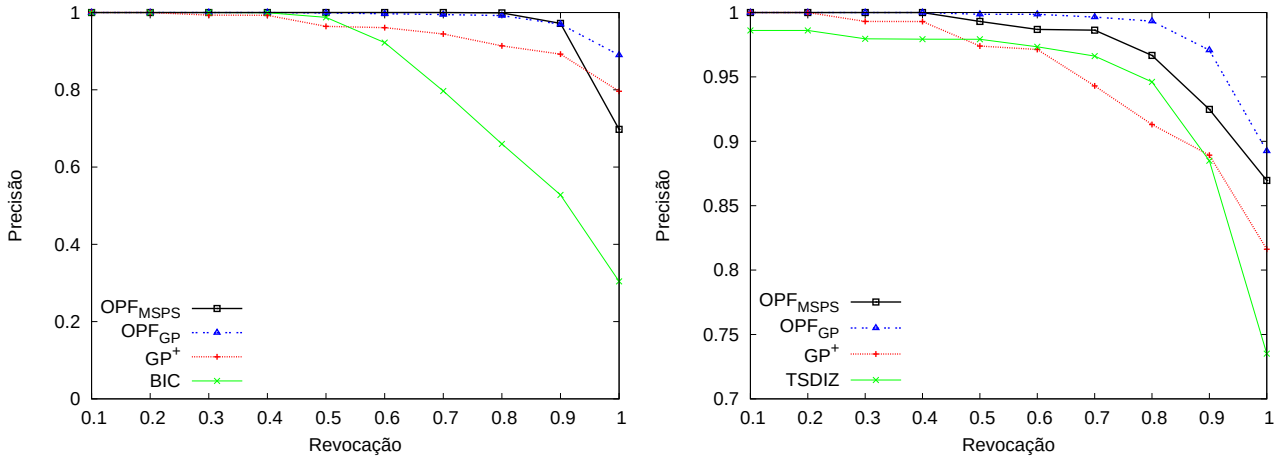


Fig. 4.16: Curva $P \times R$ média nas bases (a) ETH-80 e (b) MPEG7 após a oitava iteração.

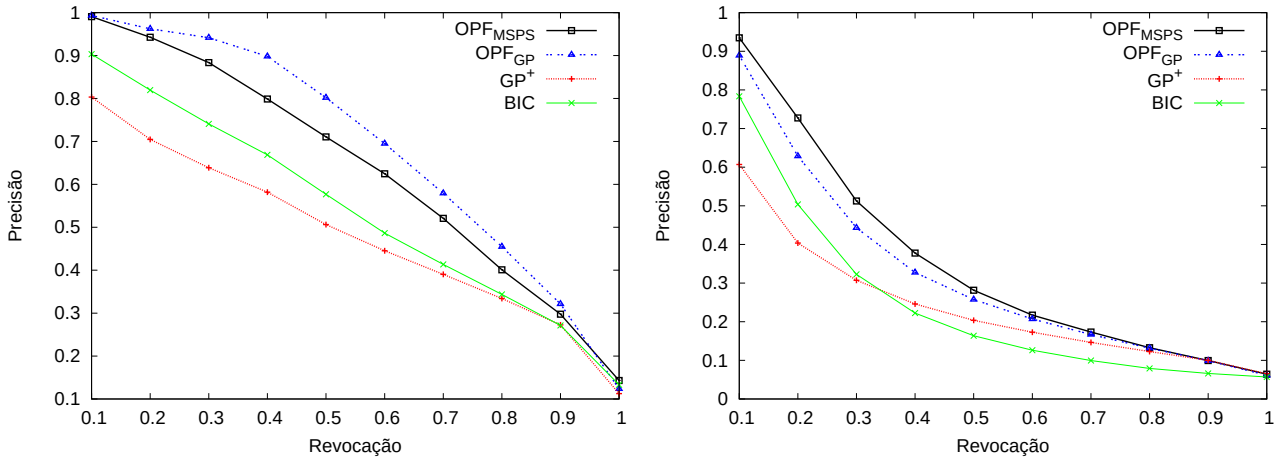


Fig. 4.17: Curva $P \times R$ média nas bases (a) MSRCORID e (b) Pascal após a terceira iteração.

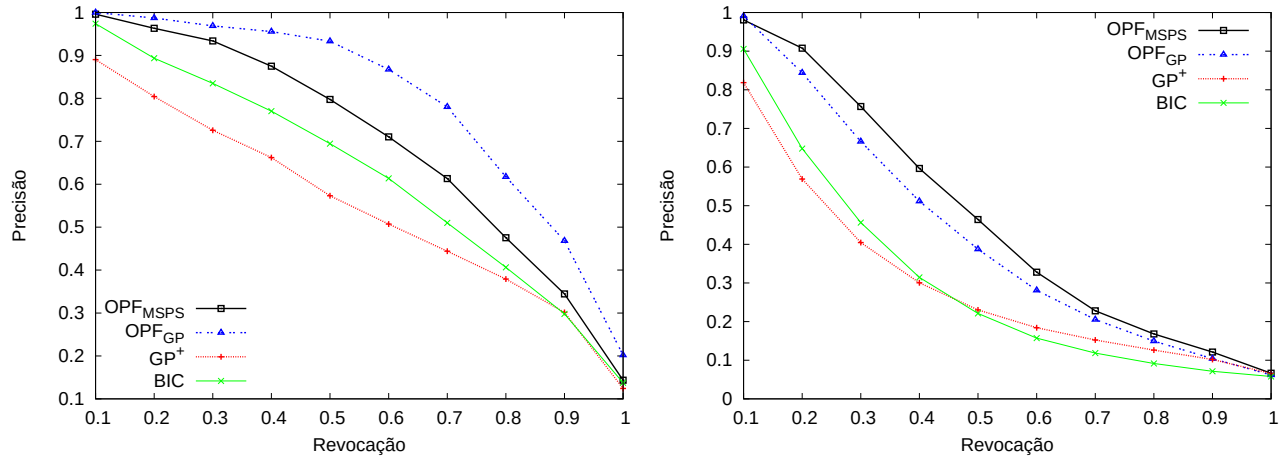


Fig. 4.18: Curva $P \times R$ média nas bases (a) MSRCORID e (b) Pascal após a quinta iteração.

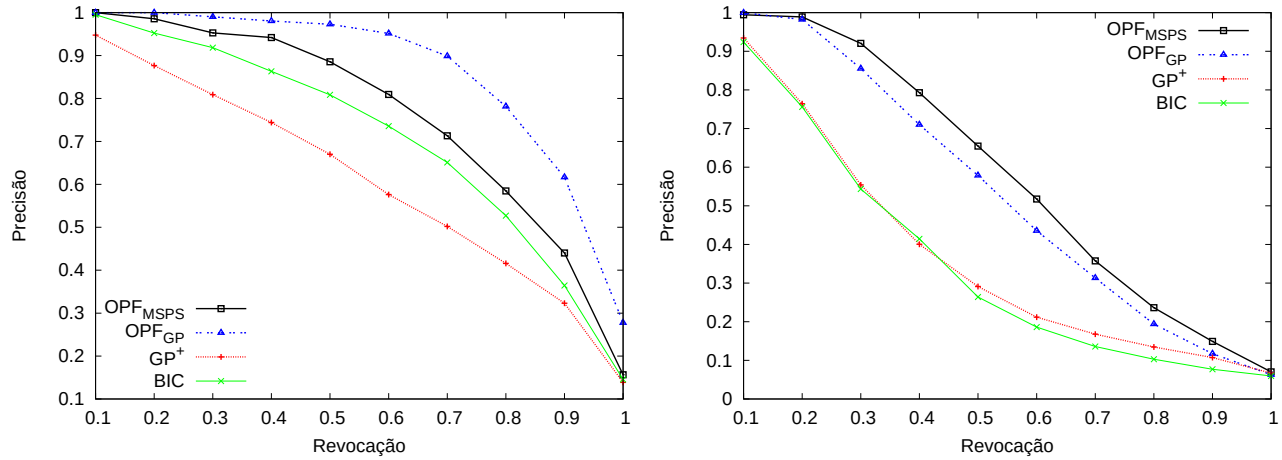
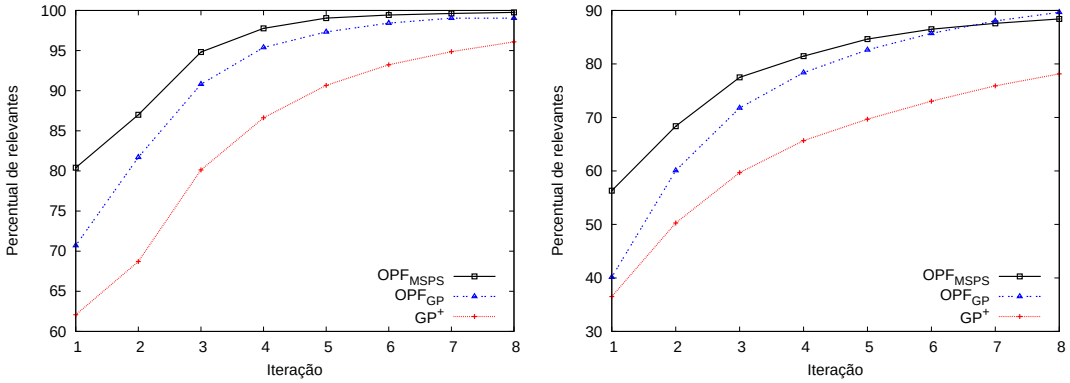
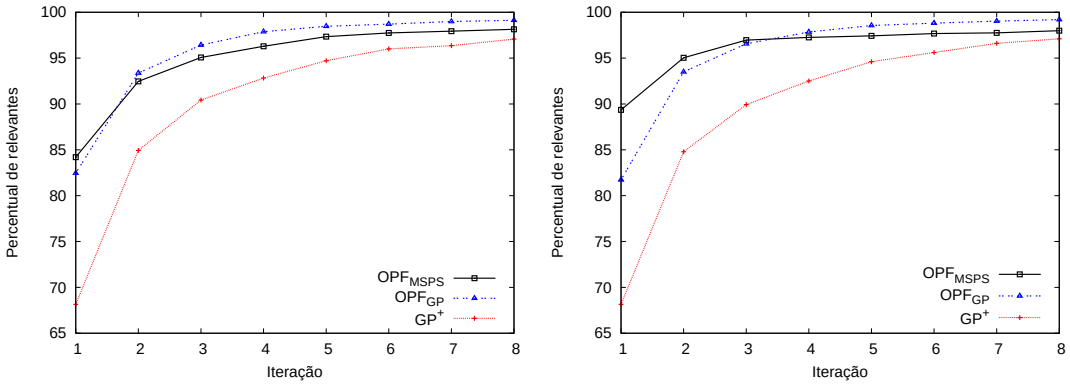


Fig. 4.19: Curva $P \times R$ média nas bases (a) MSRCORID e (b) Pascal após a oitava iteração.



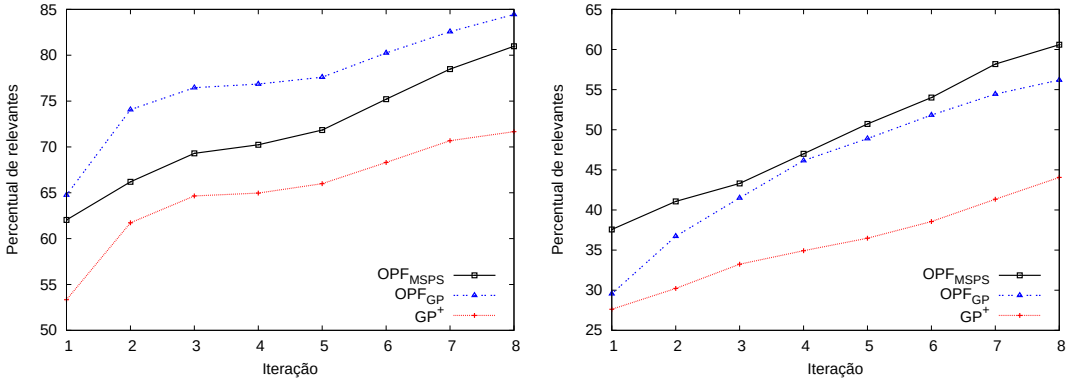
(a) (b)

Fig. 4.20: Curva $Rel \times It$ média nas bases (a) Coil-100 e (b) Corel da primeira à oitava iteração.



(a) (b)

Fig. 4.21: Curva $Rel \times It$ média nas bases (a) ETH-80 e (b) MPEG7 da primeira à oitava iteração.



(a) (b)

Fig. 4.22: Curva $Rel \times It$ média nas bases (a) MSRCORID e (b) Pascal da primeira à oitava iteração.

4.5 Resultados de $OPF_{Bi-level}$

Esta seção apresenta resultados do método de realimentação de relevância $OPF_{Bi-level}$ (Seção 3.5), que utiliza o classificador por floresta de caminhos ótimos em dois níveis de interesse. As Figuras 4.23 a 4.26 apresentam uma busca por similaridade usando a base Corel (Corel Corp.) e o descritor de imagens BIC (Stehling et al., 2002) para mostrar como a classificação dos pixels ajuda no aumento da eficácia na busca de imagens.

A Figura 4.23 é a imagem de consulta inicial e as $N = 20$ imagens mais similares de acordo com o descritor BIC são apresentadas na Figura 4.24. Se o objetivo do usuário é encontrar imagens que contenham a guarda real da Figura 4.23, é possível observar que o fundo da imagem exerce uma influência grande nos resultados obtidos. Ao definir as regiões de objeto e fundo (Figura 4.25) através da marcação de pixels, é extraído o vetor de característica desta imagem utilizando o descritor BIC apenas para a região classificada como objeto. As demais imagens da base são também segmentadas e os vetores de característica de suas regiões classificadas como objeto são extraídas. O resultado da Figura 4.26 mostra as 20 imagens mais similares à imagem de consulta inicial utilizando somente a marcação de pixels.

Fig. 4.23: Imagem de consulta inicial.

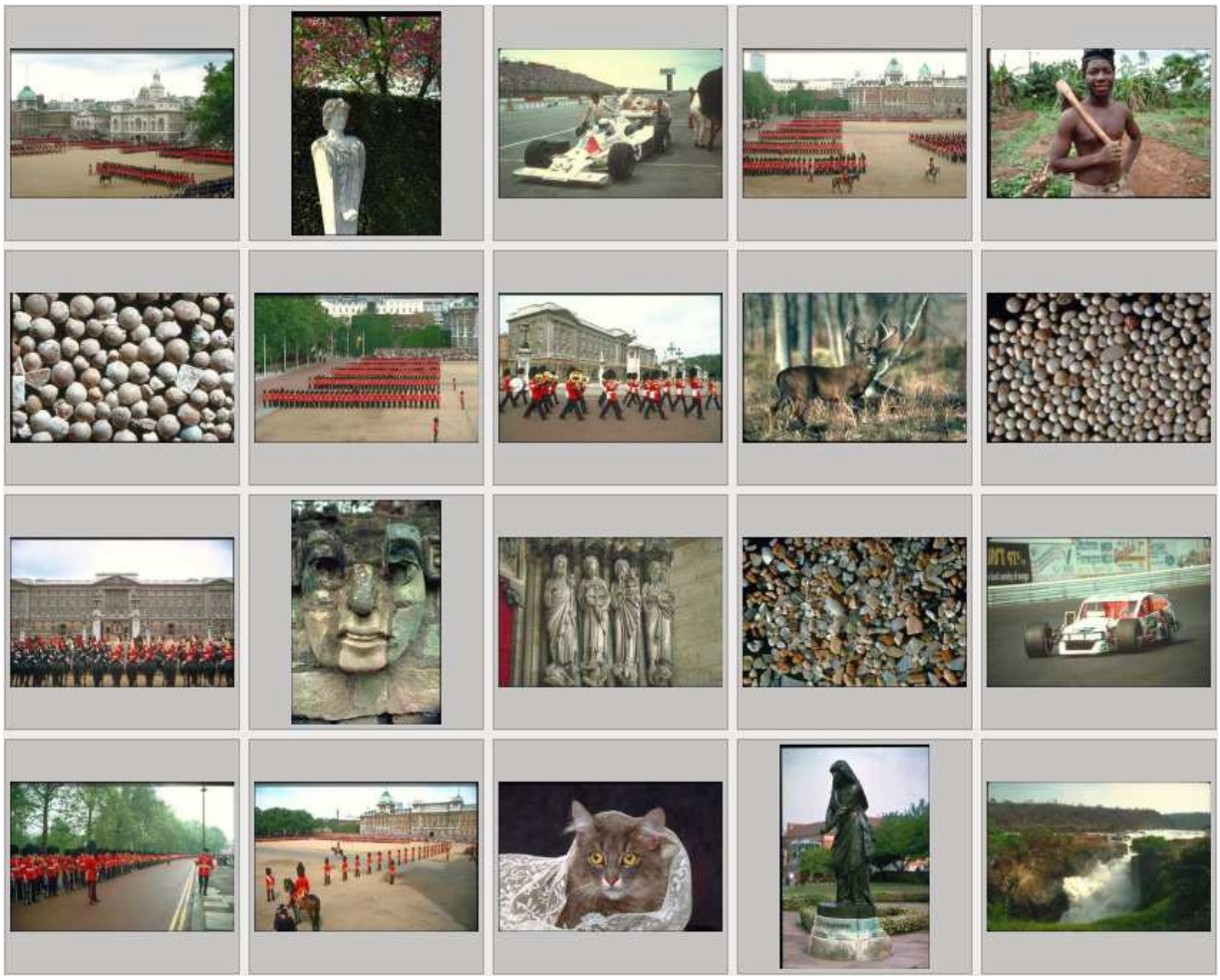


Fig. 4.24: As vinte imagens mais próximas utilizando a busca por similaridade.

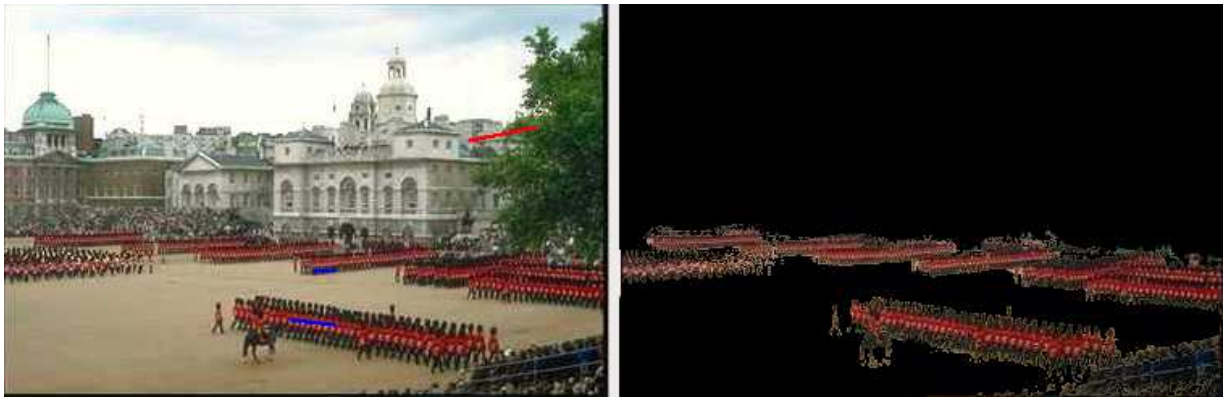


Fig. 4.25: Seleção da região de interesse.

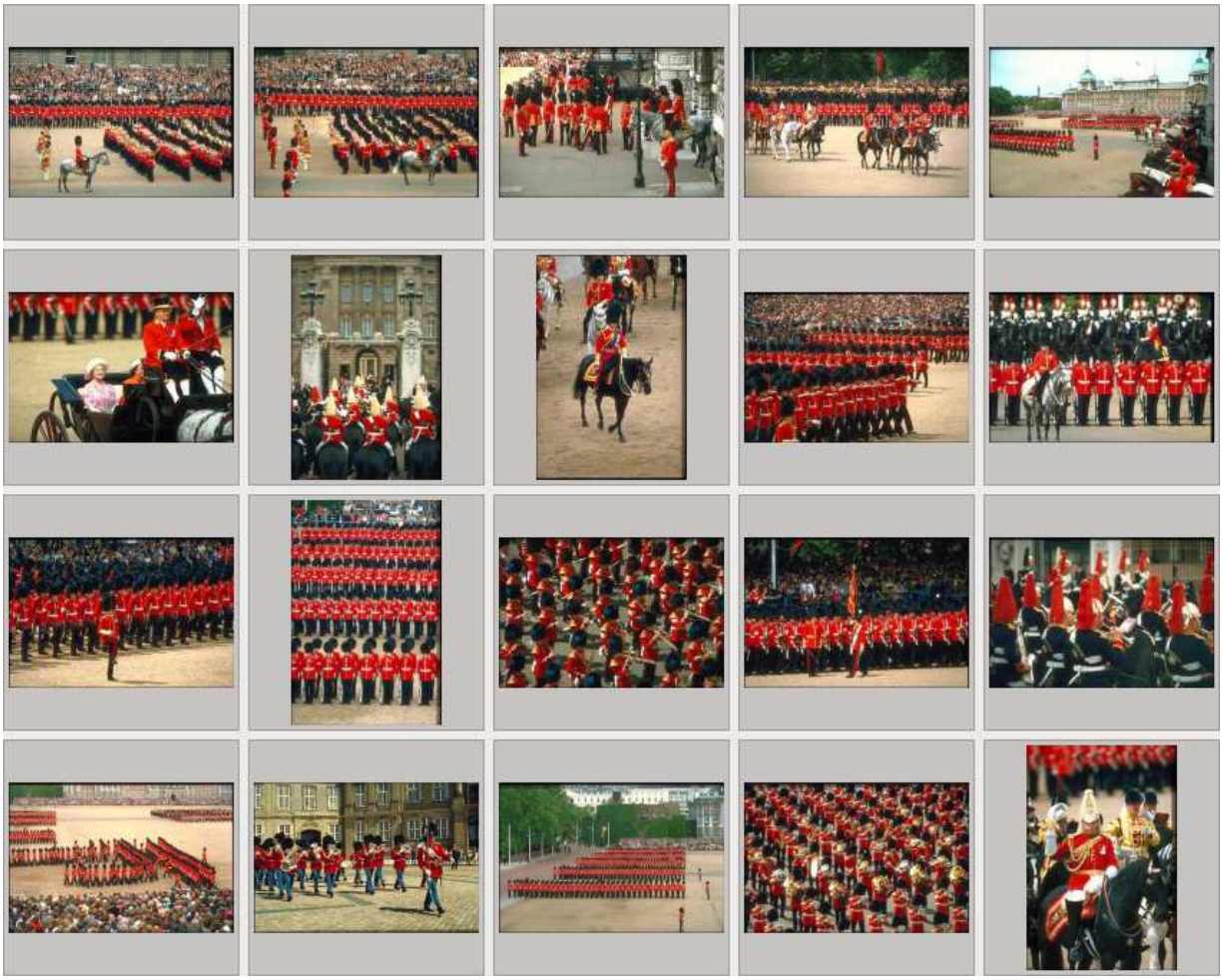


Fig. 4.26: As vinte imagens mais próximas utilizando classificação de pixels e busca por similaridade.

As Figuras 4.23 a 4.26 mostram como a classificação dos pixels em objeto e fundo melhoram o resultado de uma busca por similaridade. Nota-se que a quantidade de imagens da guarda real britânica na Figura 4.26 é maior do que na Figura 4.24. Apesar do resultado ser bastante satisfatório para este exemplo utilizando apenas a marcação de pixels, a busca de imagens não é trivial para muitos casos como pode ser visto no exemplo a seguir.

Como já exposto, o método $OPF_{Bi-Level}$ é uma nova abordagem na qual o usuário pode selecionar tanto as imagens relevantes e irrelevantes como também selecionar interativamente os objetos de interesse. Devido a essa característica, não é possível simular automaticamente o comportamento do usuário utilizando uma base de imagens previamente rotulada e segmentada, da mesma forma que foram realizados os testes anteriores (*leave-one-out*). Por isso, é apresentado a seguir um exemplo de execução do algoritmo de realimentação de relevância em dois níveis de interesse nas Figuras 4.27 a 4.33, escolhendo uma classe de imagens de difícil solução (imagens de estátua) pelos métodos de

recuperação de imagens usando a base Corel.

A Figura 4.27 mostra a consulta inicial, enquanto a Figura 4.28 apresenta as $N = 20$ imagens mais similares de acordo com o descritor BIC. Ou seja, usando somente a busca por similaridade como no exemplo anterior a fim de mostrar as imagens que seriam retornadas na primeira iteração do método $GOPF_{RF}$. Ao selecionar as regiões de objeto e fundo (Figura 4.29), a busca por similaridade retorna as imagens da Figura 4.30. Supondo que o usuário esteja interessado em imagens de estátua, verifica-se que o resultado ainda não é satisfatório conforme ocorreu no exemplo anterior. Após selecionadas as imagens relevantes, sem refazer a marcação nos pixels, é realizada a primeira iteração do algoritmo de realimentação de relevância e apresentadas novas imagens para o usuário (Figura 4.31). São então selecionadas as imagens relevantes e é realizado um ajuste nas marcações para as regiões de objeto e fundo (Figura 4.32). Após mais uma iteração do algoritmo de realimentação de relevância, é apresentado o conjunto de imagens resultante (Figura 4.33).

Fig. 4.27: Imagem de consulta inicial.

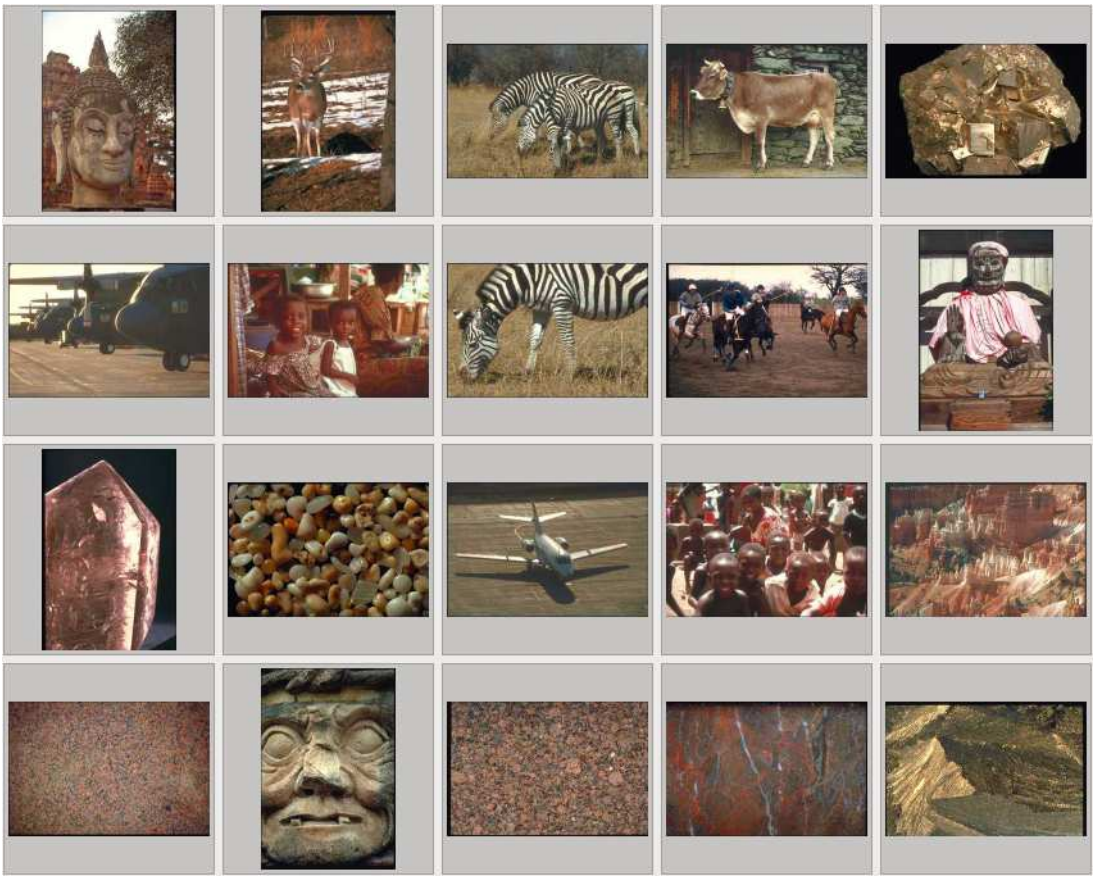


Fig. 4.28: As vinte imagens mais próximas utilizando o descritor BIC e busca por similaridade.

Fig. 4.29: Seleção da região de interesse na imagem de consulta inicial.

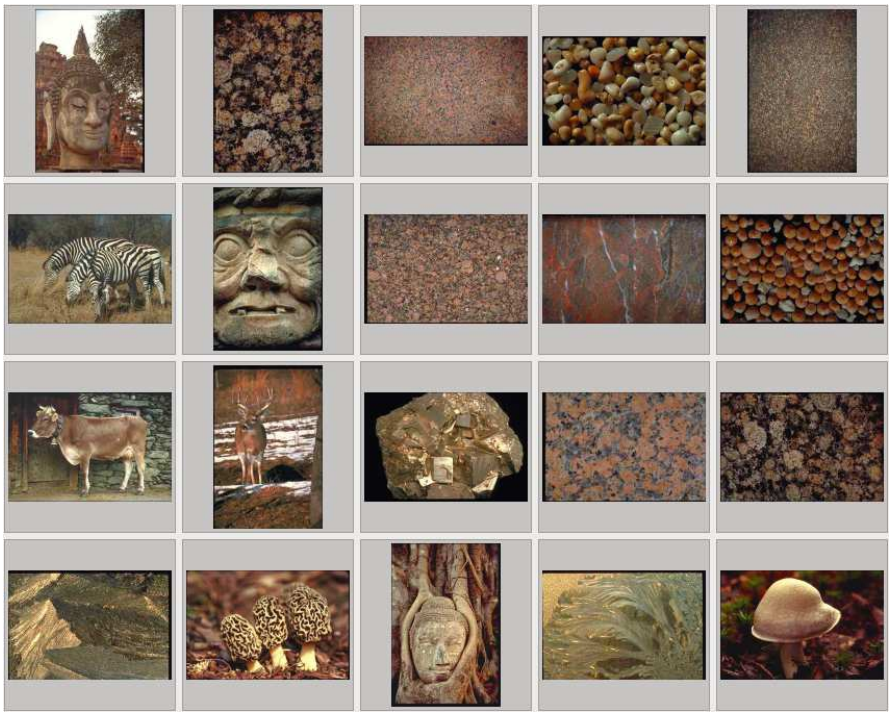


Fig. 4.30: As vinte imagens mais próximas utilizando classificação de pixels e busca por similaridade.

Fig. 4.31: Próxima iteração do método de realimentação de relevância.

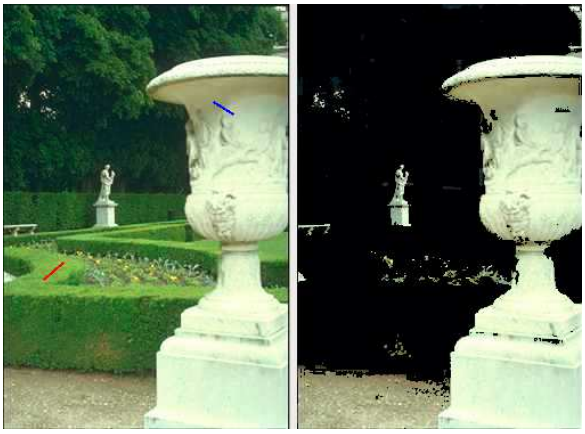


Fig. 4.32: Ajuste na seleção da região de interesse.

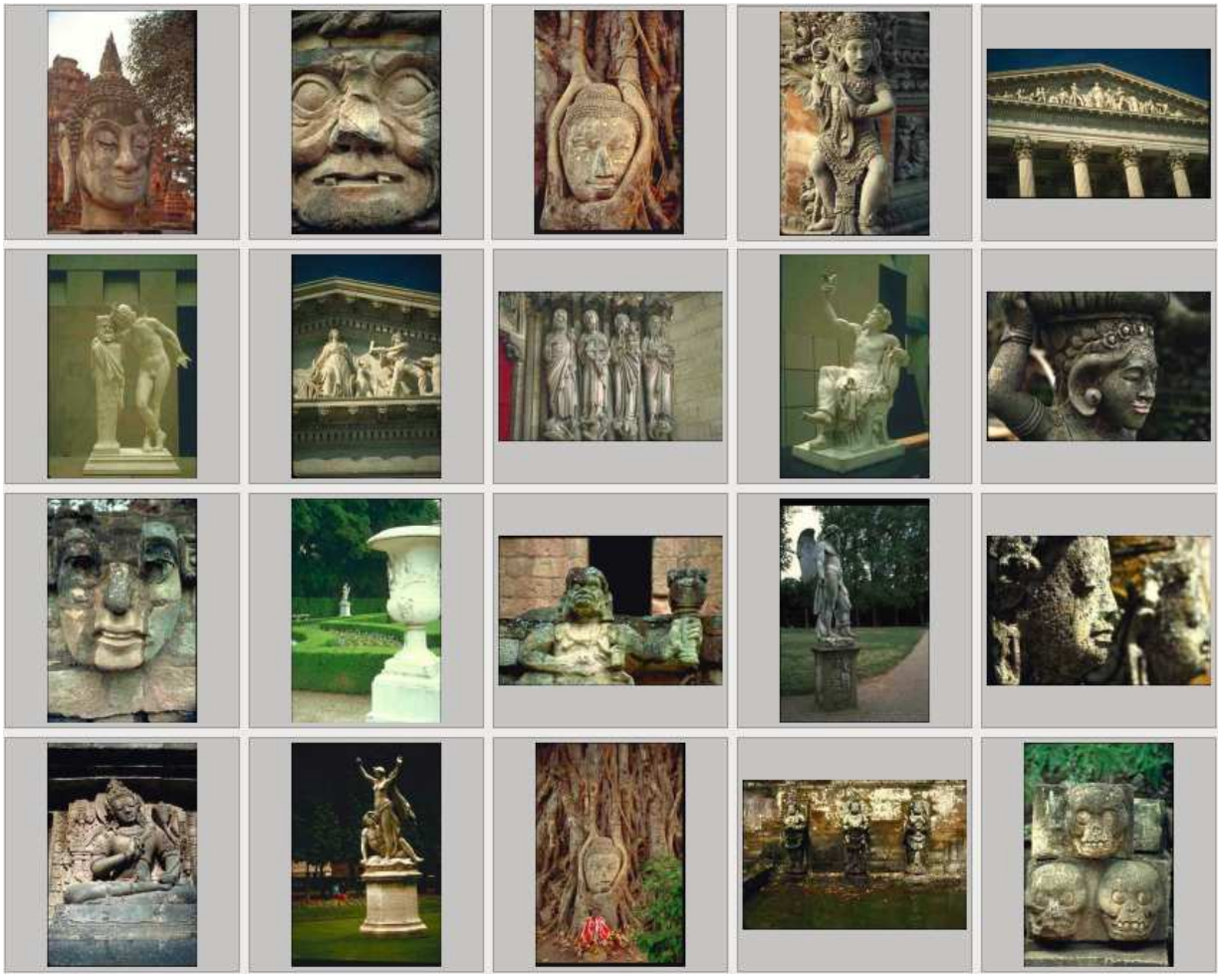


Fig. 4.33: Resultado final após duas iterações de realimentação de relevância.

Utilizando as marcações de objeto e fundo feitas para este exemplo (Figuras 4.29 e 4.32) é apresentada na Figura 4.34 a curva média $Rel \times It$ da execução usando as 46 imagens da classe "estátua" como consulta inicial através da técnica de validação cruzada *leave-one-out*. São mostrados os resultados de precisão para as iterações 1 a 8, usando $N = 20$ imagens por iteração e os rótulos definidos pela base Corel. No mesmo gráfico também é apresentado o resultado da curva média $Rel \times It$ para o método $GOPF_{RF}$ utilizando o descritor BIC nas imagens inteiras. A classe da imagem de consulta (Figura 4.27) possui 43 imagens de um total de 3.906.

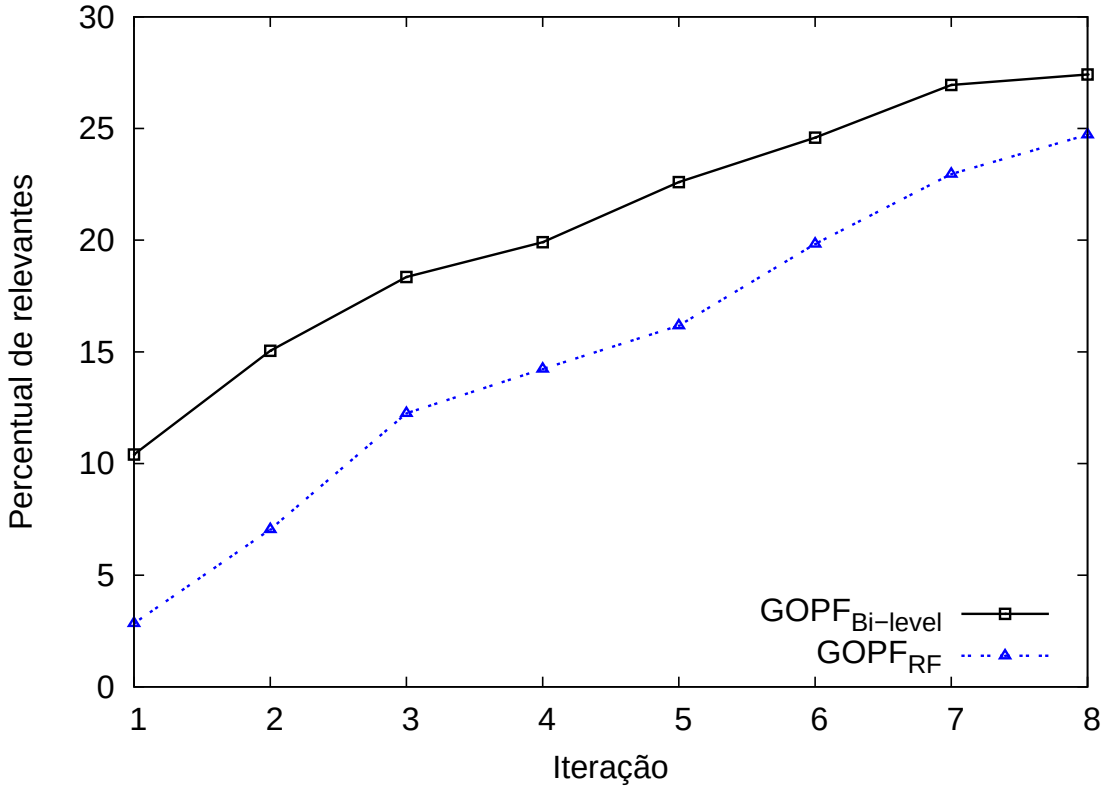


Fig. 4.34: Curva média $Rel \times It$ na base Corel para as imagens da classe "estátua".

A Figura 4.35 mostra a curva $Rel \times It$ usando apenas a imagem da Figura 4.27 como entrada. A curva, mostrando o número de imagens obtidas nas iterações 1 a 8, também foi gerada usando $N = 20$ imagens por iteração, os rótulos definidos pela base Corel e as marcações de pixels das Figuras 4.29 e 4.32.

Nota-se, observando as Figuras 4.34 e 4.35, que o método de realimentação de relevância usando o classificador OPF em dois níveis de interesse consegue obter uma eficácia melhor do que aquela alcançada pelo método $GOPF_{RF}$, mostrando que a marcação de objeto e fundo melhora a eficácia do método de realimentação de relevância baseado em floresta de caminhos ótimos.

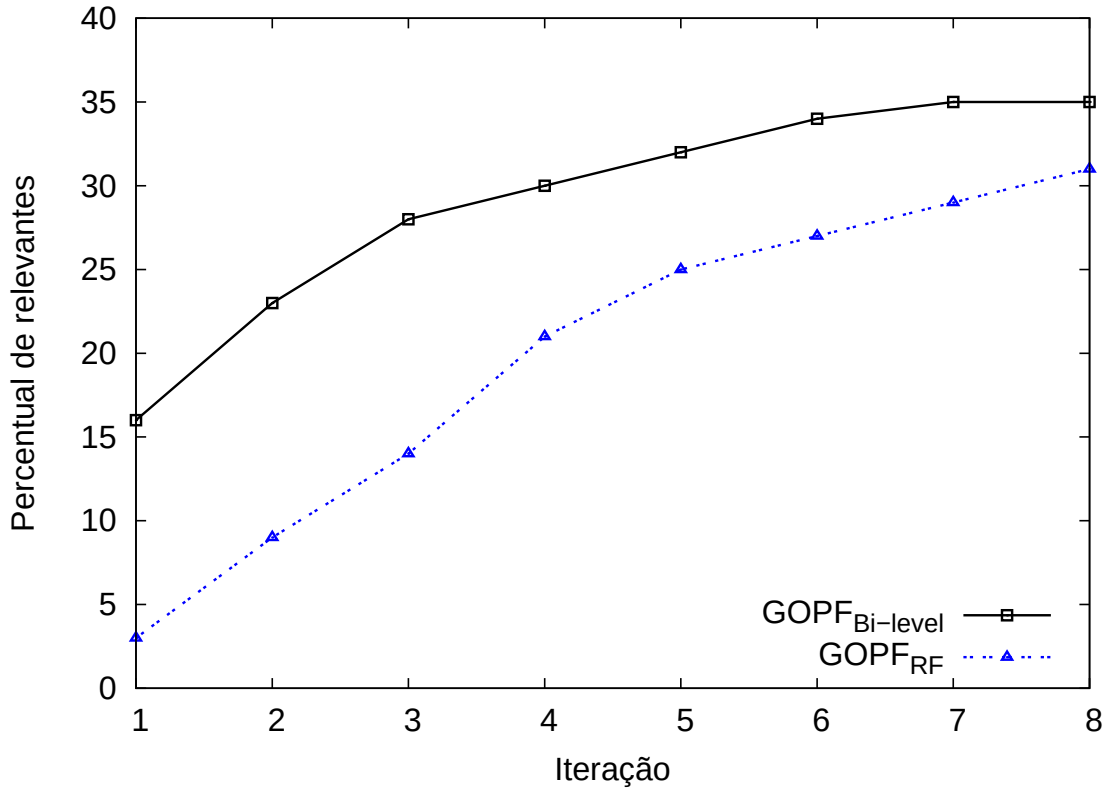


Fig. 4.35: Curva $Rel \times It$ na base Corel utilizando a Figura 4.27 como consulta inicial.

As Figuras 4.36 a 4.38 mostram as curvas do número de relevantes por iteração para as bases MSRCORID, Pascal e Caltech. A execução nestas outras três bases de imagens foram realizadas conforme apresentado para a Figura 4.34. A Figura 4.36 mostra a média da execução para todas as imagens de vaca na base MSRCORID e a Figura 4.37 mostra a média para todas as imagens de ovelha na base Pascal. Conforme apresentado na Seção 3.5, a marcação do usuário interfere no resultado da busca. A Figura 4.38 mostra a curva média obtida para todas as imagens de avião na base Caltech, mostrando uma marcação bem feita pelo Usuário 1 e outra marcação mal sucedida feita pelo Usuário 2.

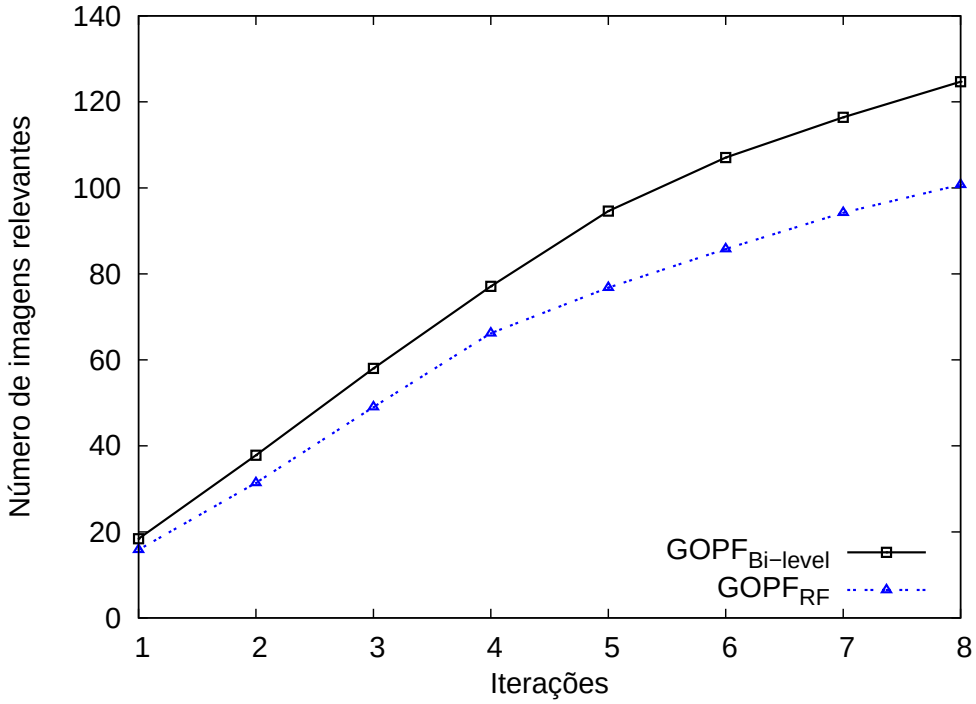


Fig. 4.36: Curva média de relevantes $\times$ iteração na base MSRCORID para as imagens da classe "vaca".

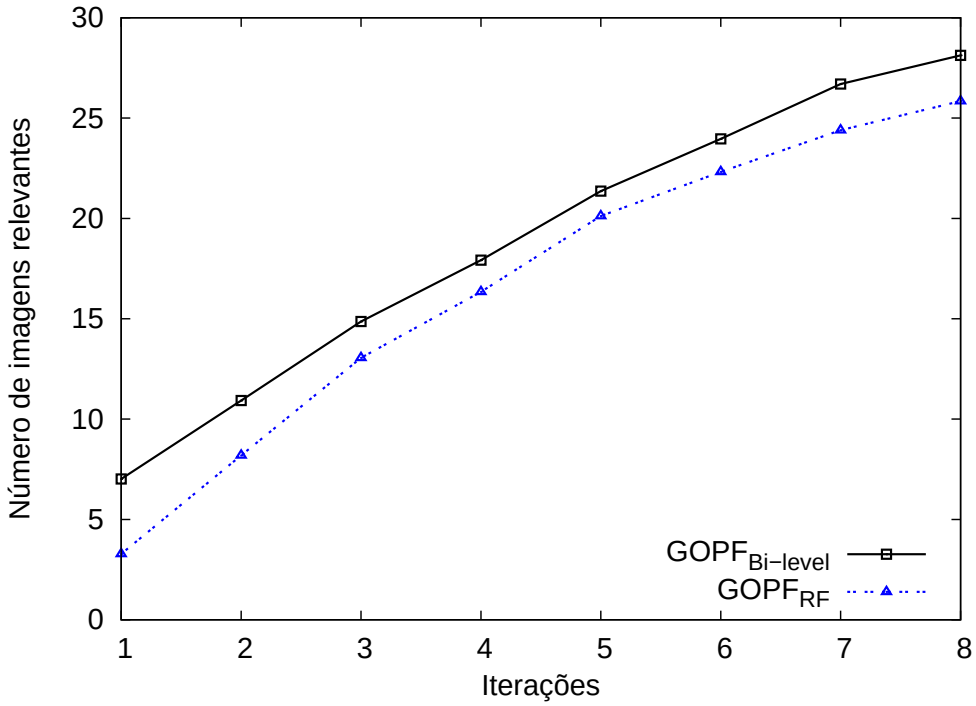


Fig. 4.37: Curva média de relevantes $\times$ iteração na base Pascal para as imagens da classe "ovelha".

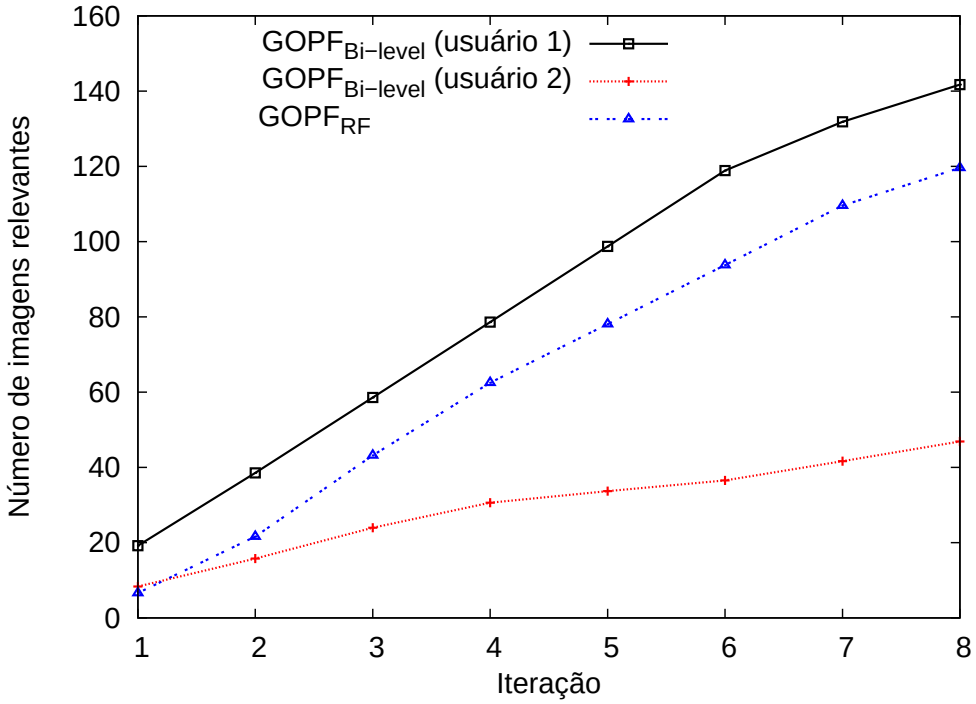


Fig. 4.38: Curva média de relevantes $\times$ iteração na base Caltech para as imagens da classe “avião”.

Esta Seção mostrou um exemplo simples de execução para servir como teste de conceito de um método em desenvolvimento, além de apresentar os resultados obtidos por esta técnica em quatro bases de imagem. O próximo capítulo conclui esta tese com considerações sobre os métodos apresentados e as perspectivas futuras em relação a esses métodos.

Capítulo 5

Conclusão

Esta tese teve o objetivo de propor novos métodos dentro do paradigma de realimentação de relevância através do uso do classificador baseado em floresta de caminhos ótimos. Neste contexto, este classificador foi pela primeira vez utilizado para testar conjuntos de treinamento pequenos, conforme exigido pela técnica de aprendizado por realimentação de relevância. Foi mostrado que o uso do classificador OPF para recuperação de imagens por conteúdo é eficiente e eficaz, necessitando de poucas iterações para apresentar resultados desejados. Esta tese também contribui para a área de CBIR ao definir dois paradigmas distintos de acordo com a ordenação das imagens apresentadas a cada iteração. Adicionalmente, foi definido um novo método para combinação de descritores utilizando o método de otimização MSPS (*Multi-Scale Parameter Search*). Por fim, é apresentada uma nova metodologia de realimentação de relevância em dois níveis de interesse usando o classificador OPF.

Os métodos propostos criam, de acordo com as imagens rotuladas previamente pelo usuário ao longo das iterações, um grafo completo onde os arcos são definidos pela similaridade entre as imagens. Este grafo é usado para gerar uma MST (*Minimum Spanning Tree* ou árvore espalhada mínima) e os elementos adjacentes com diferentes rótulos (relevante e irrelevante) definem os protótipos. Com isso, são criadas as florestas de caminhos ótimos que definem o classificador OPF. Cada imagem restante da base é então classificada como relevante ou não e, ao final do processo, somente as imagens classificadas como relevantes são ordenadas e apresentadas ao usuário. Para retornar primeiramente as imagens que são as mais prováveis de serem relevantes de acordo com as especificações do usuário foi utilizada uma métrica baseada na distância média entre os protótipos relevantes e irrelevantes definidos pelo classificador OPF.

O Capítulo 3 apresentou cada um dos métodos desenvolvidos nesta tese para recuperação de imagens usando a técnica de realimentação de relevância baseada no classificador OPF, enquanto no Capítulo 4, foram apresentados os resultados da aplicação destes métodos em sete bases de imagens (Caltech 101, Coil-100, Corel, ETH-80, MPEG7, MSRCORID e PASCAL) que representam diferen-

tes desafios na área de CBIR, de forma a medir a eficiência e eficácia dos métodos aqui desenvolvidos. Os resultados foram comparados com os valores obtidos por métodos considerados como estado-da-arte em realimentação de relevância, SVM_{AL} e QPM, sem combinação de descritores, e GP^{+} para os métodos usando combinação de descritores.

Esta tese denominou os dois paradigmas distintos de aprendizagem por realimentação de relevância em relação às imagens retornadas como guloso e planejado. No paradigma guloso, a cada iteração tenta-se retornar sempre as imagens que o usuário considera mais relevantes, sendo este o paradigma mais utilizado em CBIR. Já no paradigma planejado, o usuário estabelece em quantas iterações o sistema deverá aprender, retornando imagens mais informativas para o classificador a cada iteração. Métodos baseados em SVM utilizam apenas o paradigma planejado, enquanto o método aqui apresentado usando o classificador OPF pode usar ambos os paradigmas de aprendizagem. São apresentados nesta tese os dois métodos de realimentação de relevância usando OPF, um utilizando o paradigma guloso ($GOPF_{RF}$) e o outro o paradigma planejado ($POPF_{RF}$).

Embora o número de iterações necessárias para o aprendizado pareça ser maior na abordagem planejada, isso não é necessariamente verdade, como mostrado na Seção 4.3. A técnica usando o paradigma planejado garante um ganho em eficácia sobre o método usando o paradigma guloso, como demonstrado ao comparar os resultados dos métodos $POPF_{RF}$ e $GOPF_{RF}$ (Figuras 4.4 a 4.10). Esta tese mostra que quando o número de imagens da classe pesquisada é muito pequena em relação à quantidade total de imagens da base de dados é mais eficiente retornar somente aquelas classificadas como candidatas a relevante dentre as mais difíceis de classificar, devido ao grande número de falsos positivos (Seção 3.2).

De fato, este ganho provavelmente será sempre verificado ao se comparar um mesmo método usando ambos os paradigmas, já que é mais eficaz retornar imagens que auxiliam o aprendizado do classificador. Por isso, é importante observar que o método $GOPF_{RF}$ ao usar o paradigma guloso, para poucas iterações, consegue ser mais eficaz do que o método planejado SVM_{AL}. Os resultados obtidos indicam que os métodos de realimentação de relevância baseados em floresta de caminhos ótimos ($GOPF_{RF}$ e $POPF_{RF}$) necessitam de poucas iterações para obter um resultado desejado e superam em eficácia os métodos de referência (QEX e SVM_{AL}) em todas as bases de dados testadas. Embora o método planejado tenha obtido uma maior eficácia em todas as bases testadas, a escolha de uso entre $GOPF_{RF}$ e $POPF_{RF}$ depende da aplicação. Para um usuário comum, por exemplo, é mais agradável visualizar imagens cada vez mais similares ao que ele deseja.

Também é importante notar que a perda de imagens relevantes pelo classificador OPF pode ser considerada insignificante (cerca de 1,5%). Assim, o método desenvolvido nesta tese fornece uma solução em tempo real para aplicações interativas de recuperação de imagem, em média 52 vezes mais rápido do que SVM_{AL}, conforme as Tabelas 4.1 e 4.2.

Conforme citado na Seção 4.3, não foi utilizada nenhuma estrutura de indexação para a geração dos resultados nas bases testadas. O fato do classificador OPF utilizar um grafo completo para o treinamento poderia sugerir que fosse necessário utilizar estruturas de dados de indexação para suportar o uso de bases grades. No entanto, o tamanho do grafo (quantidade de nós) é relacionado com a quantidade N de imagens apresentadas ao usuário a cada iteração e ao número I de iterações ($N \times I$), sendo normalmente uma quantidade muito pequena. Desta forma, o classificador OPF não necessita do uso de técnicas de indexação para seu treinamento. Deve-se ter presente que técnicas de indexação podem ser empregadas juntamente com o método de realimentação de relevância para aumentar ainda mais a eficiência da busca de imagens em bases muito grandes.

Com relação a utilização de combinação de descritores, esta tese apresenta o método OPF_{MSPS} como descrito na Seção3.3. A combinação de descritores pode ser aplicada aos métodos $GOPF_{RF}$ e $POPF_{RF}$ para utilizar diferentes características da imagem (cor e textura, por exemplo). Para isto, uma função de combinação deve definir os valores usados no cálculo de similaridade entre duas imagens e, consequentemente, gerar os valores dos arcos na criação do classificador OPF, conforme comentado a seguir.

Foram utilizadas duas técnicas de combinação de descritores juntamente com os métodos de realimentação de relevância baseados em OPF a fim de melhorar a eficácia do processo de aprendizagem. Na primeira, o método de otimização chamado MSPS é utilizado pela primeira vez para a combinação de descritores (OPF_{MSPS}), enquanto na segunda é utilizada uma técnica consolidada baseada em programação genética (OPF_{GP}). Percebe-se que a combinação de descritores usando tanto OPF_{MSPS} quanto OPF_{GP} é bastante eficaz. A escolha entre os dois métodos depende do tipo de busca a ser realizada. Se o projetista do sistema conhece aproximadamente qual função de combinação é a mais adequada para o problema, OPF_{MSPS} pode ser mais adequado. Se nenhuma relação entre os diferentes descritores é conhecida ou se for necessário combinar muitos descritores, OPF_{GP} pode ser a melhor escolha. OPF_{MSPS} foi mais rápido do que OPF_{GP} nos testes realizados, embora seja difícil comparar os tempos de execução de ambos os métodos, já que esse tempo varia de acordo com o número de descritores a serem combinados. A quantidade de escalas no MSPS também pode influenciar neste tempo, mas testes indicaram que quanto menor o número de escalas, mais iterações são necessárias para se encontrar os melhores parâmetros da função de combinação. Para o caso do método baseado em programação genética, o número de descritores não influencia o tempo de execução, mas sim a quantidade de indivíduos e gerações envolvidos para evolução dos indivíduos.

Outro fator a considerar é que OPF_{GP} é mais flexível do que OPF_{MSPS}, podendo gerar funções de combinação bastante diversificadas. Por outro lado, os métodos baseados em programação genética são sensíveis aos parâmetros iniciais, necessitando que se faça um estudo paramétrico para encontrar o melhor conjunto de parâmetros para cada caso. No método usando MSPS desta tese, foi

utilizada uma mesma função e o mesmo conjunto de escalas para todos os resultados gerados. Para um problema específico, seria possível encontrar uma função mais adequada ao problema em questão a fim de melhorar a eficácia do método.

Por último, uma nova metodologia de realimentação de relevância em dois níveis de interesse ($OPF_{Bi-Level}$) foi apresentada. Nela, o usuário não só marca as imagens relevantes e irrelevantes como também pode selecionar nas imagens os pixels que ele considera importantes para a consulta. O classificador por floresta de caminhos ótimos também é utilizado para a seleção das regiões de imagens, através de um método interativo de classificação dos pixels. O usuário marca na imagem de consulta inicial quais objetos ele considera relevantes e o sistema classifica em todas as outras imagens da base quais pixels deverão ser considerados e quais podem ser ignorados (fundo da imagem ou objetos irrelevantes). Desta forma, somente são extraídas as características das regiões consideradas de interesse para definir a similaridade entre elas. Durante o processo de realimentação de relevância o usuário pode tanto definir quais as imagens são relevantes ou não (abstração por objeto), quanto selecionar o que foi erroneamente classificado como objeto ou fundo para qualquer das imagens (abstração por pixel). O exemplo apresentado (Figuras 4.27 a 4.35) mostra que esta nova metodologia parece ser muito promissora, conseguindo obter uma eficácia melhor do que a alcançada somente com a seleção de imagens relevantes ou não durante a aprendizagem por realimentação de relevância.

O problema deste método é seu custo computacional. Para resolver este problema, foi adotado o uso de *thumbnails*, onde os pixels das versões reduzidas das imagens são classificados como objeto ou fundo, de acordo com as florestas de caminhos ótimos criadas usando o conjunto de pixels rotulados pelo usuário. As regiões formadas pelos pixels classificados como objeto são mapeadas na imagem de tamanho original para calcular os vetores de característica, reduzindo a quantidade de pixels a serem classificados como objeto e fundo.

Para concluir é importante novamente frisar que a utilização da classificação de floresta de caminhos ótimos para recuperação de imagens mostrou ser uma abordagem muito promissora sendo bastante rápida e eficaz, necessitando de poucas iterações para apresentar resultados satisfatórios. Dentre os métodos propostos, é possível utilizar tanto a abordagem gulosa quanto a planejada com a inclusão ou não de combinação de descritores, sem a necessidade de modificações no algoritmo. O uso do classificador OPF para selecionar os pixels de objeto e fundo ajuda a aumentar ainda mais a eficácia do método de realimentação de relevância.

Como trabalho futuro imediato, pretende-se utilizar um método[1] para acelerar o processo de classificação dos pixels na técnica de realimentação de relevância em dois níveis de interesse, selecionando um número mais reduzido de amostras de treinamento mas mantendo a acurácia na classi-

[1] C.C.C. Fernández. "Novos Algoritmos de Aprendizado para Classificação de Padrões Utilizando Floresta de Caminhos Ótimos". Dissertação de Mestrado, Unicamp, 2011.

ficação das regiões de objeto e fundo. Técnicas de processamento massivo paralelo (como GPUs) também deverão ser empregadas para acelerar o processo de classificação. Também estuda-se o uso de descritores locais como SIFT (*Scale-Invariant Feature Transform*) e SURF (*Speeded Up Robust Features*) no método $OPF_{Bi-Level}$. Para isso, somente os pontos de interesse são classificados a fim de identificar se estão na região de um objeto, em vez de classificar todos os pixels da imagem para depois extrair as características da região de objeto. Em relação à combinação de descritores, será avaliado o uso de um outro método denominado $GP^{\pm}$ (Ferreira et al., 2011) para substituir o algoritmo utilizado em OPF_{GP} e a definição de funções de combinação mais eficientes para o MSPS. Por fim, pretende-se realizar experiências com usuários, quantificando também os resultados através de experimentos de campo.

Referências Bibliográficas

Andaló, F. A., Miranda, P. A. V., Torres, R. S. e Falcão, A. X. Shape feature extraction and description based on tensor scale. *Pattern Recognition*, 43(1):26–36, 2010. ISSN 0031-3203.

Anh, N. D., Bao, P. T., Nam, B. N. e Hoang, N. H. A new CBIR system using sift combined with neural network and graph-based segmentation. In *International Conference on Intelligent Information and Database Systems*, ACIIDS'10, pages 294–301, Berlin, Heidelberg, 2010. Springer-Verlag.

Arevalillo-Herráez, M., Ferri, F. J. e Domingo, J. A naive relevance feedback model for content-based image retrieval using multiple similarity measures. *Pattern Recognition*, 43(3):619–629, 2010. ISSN 0031-3203.

Barkowsky, T., Bertel, S., Engel, D. e Freksa, C. Design of an architecture for reasoning with mental images, 2003. International Workshop on Spatial and Visual Components in Mental Reasoning about Large-Scale Spaces.

Bartolini, I., Ciaccia, P. e Patella, M. Query processing issues in region-based image databases. *Knowl. Inf. Syst.*, 25:389–420, November 2010. ISSN 0219-1377.

Bay, H., Tuytelaars, T. e Gool, L. V. Surf: Speeded up robust features. In *Computer Vision and Image Understanding (CVIU)*, volume 110, pages 346–359, 2008.

Berk, T., Brownston, L. e Kaufman, A. A new color-naming system for graphics languages. In *IEEE Computer Graphics and Applications*, volume 2, pages 37–44. IEEE, 1982.

Boujnane, L. e Bloore, P. TinEye, PixID, Piximilar: advanced image software by Idée Inc. http://www.tineye.com/, 2009.

Cai, D., He, X., Li, Z., Ma, W. Y. e Wen, J. R. Hierarchical clustering of WWW image search results using visual, textual and link information. In *ACM International Conference on Multimedia*, pages 952–959, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: http://doi.acm.org/10.1145/1027527.1027747.

ÇarkacIoglu, A. e Yarman-Vural, F. SASI: a generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615 – 2633, 2003. ISSN 0031-3203.

Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M. e Malik, J. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*, pages 509–516. Springer, 1999.

Carson, C., Belongie, S., Greenspan, H. e Malik, J. Blobworld: image segmentation using expectation-maximization and its application to image querying. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1026–1038, 2002. ISSN 0162-8828.

Churchland, P. S., Ramachandran, V. S. e Sejnowski, T. J. A critique of pure vision. *Largescale neuronal theories of the brain*, pages 23–60, 1994.

Ciaccia, P., Patella, M. e Zezula, P. M-tree: an efficient access method for similarity search in metric spaces. In *VLDB International Conference*, pages 426–435, 1997.

Corel Corp. Corel stock photo library 2. Ontario, Canada.

Cormen, T., C., Leiserson. e Rivest, R. *Introduction to Algorithms*. MIT, 1990.

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V. e Yianilos, P. N. The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 1(9):20–37, January 2000.

Croft, D. e Thagard, P. Dynamic imagery: A computational model of motion and visual analogy. In *Model-Based Reasoning: Science, Technology, & Values*, pages 259–274. Kluwer Academic: Plenum Publishers, 2002.

Datta, R., Joshi, D., Li, J. e Wang, J. Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008. ISSN 0360-0300.

Davies, J., Goel, A. K. e Nersessian, N. J. A computational model of visual analogies in design. *Cognitive Systems Research*, 10(3):204–215, 2009. Special Issue on Analogies - Integrating Cognitive Abilities.

Dorairaj, R. e Namuduri, K. Compact combination of mpeg-7 color and texture descriptors for image retrieval. *Conference Record of the Thirty-Eighth Asilomar Conference on Signals*, 1(38):387–391, 2004.

Doulamis, N. e Doulamis, A. Evaluation of relevance feedback schemes in content-based in retrieval systems. *Signal Processing: Image Communication*, 21(4):334–357, 2006.

Draper, B., Baek, K. e Boody, J. Implementing the expert object recognition pathway. *Machine Vision and Applications*, 16:27–32, 2004. ISSN 0932-8092.

Drumond, T. F. e Magalhães, L. P. Estudo e implementação de descritores de cor e forma para sistemas CBIR. *Congresso Interno de Iniciação Científica da UNICAMP*, pages 352–353, 2010.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. e Zisserman, A. The pascal visual object classes challenge 2010 (voc2010). http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/index.html.

Falcão, A. X., Stolfi, J. e Lotufo, R. A. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. e Equitz, W. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, July 1994. ISSN 0925-9902.

Fan, W., Fox, E. A., Pathak, P. e Wu, H. The effects of fitness functions on genetic programming-based ranking discovery for web search. *Journal of the American Society for Information Science and Technology*, 55:2004, 2004.

Fei-Fei, L., Fergus, R. e Perona, P. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004, Workshop on Generative-Model*. Based Vision, 2004.

Felsen, G. e Dan, Y. A natural approach to studying vision. *Nature Neuroscience*, 8(12):1643–1646, 2005.

Feng, H. e Chua, T. S. A bootstrapping approach to annotating large image collection. In *ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '03, pages 55–62, New York, NY, USA, 2003. ACM. ISBN 1-58113-778-8.

Feng, H., Shi, R. e Chua, T. S. A bootstrapping framework for annotating and retrieving www images. In *ACM International Conference on Multimedia*, pages 960–967, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: http://doi.acm.org/10.1145/1027527.1027748.

Fernando, B., Fromont, E., Muselet, D. e Sebban, M. Supervised learning of gaussian mixture models for visual vocabulary generation. *Pattern Recognition*, In Press, 2011. ISSN 0031-3203.

Ferreira, C. D., Santos, J. A., Torres, R. S., Gonçalves, M. A., Rezende, R. C. e Fan, W. Relevance feedback based on genetic programming for image retrieval. *Pattern Recognition Letters*, 32(1):27 – 37, 2011. ISSN 0167-8655.

Gelasca, E. D., Guzman, J. D., Gauglitz, S., Ghosh, P., Xu, J., Moxley, E., Rahimi, A. M., Bi, Z. e Manjunath, B. S. Cortina: Searching a 10 million + images database. Technical report, Sep 2007.

Gevers, T. e Smeulders, A. W. M. Pictoseek: combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102 –119, 2000. ISSN 1057-7149.

Giacinto, G. e Roli, F. Bayesian relevance feedback for content-based image retrieval. *Pattern Recognition*, 37(7):1499–1508, 2004. ISSN 0031-3203.

Glasgow, J. I. e Papadias, D. Computational imagery. *Cognitive Science*, 16(3):355–394, 1992.

Gorder, P. F. Computer vision, inspired by the human brain. *Computing in Science and Engineering*, 10:6–11, March 2008. ISSN 1521-9615.

Gupta, A. e Jain, R. Visual information retrieval. *Communications of the ACM*, 40:70–79, May 1997. ISSN 0001-0782.

Hoi, S. C. H. e Lyu, M. R. A semi-supervised active learning framework for image retrieval. volume 2, pages 302–309, jun. 2005.

Hoi, S. C. H., Liu, W. e Chang, S. F. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications and Applications*, 6(3):1–26, 2010. ISSN 1551-6857.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W. J. e Zabih, R. Image indexing using color correlograms. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 762–768, 1997. ISSN 1063-6919.

Huang, W., Gao, Y. e Chan, K. L. A review of region-based image retrieval. *Journal of Signal Processing Systems*, 59:143–161, May 2010. ISSN 1939-8018.

Iqbal, Q. e Aggarwal, J. K. Cires: a system for content-based retrieval in digital image libraries. In *in Invited Session on Content-based Image Retrieval: Techniques and Applications, 7 th International Conference on Control Automation, Robotics and Vision (ICARCV*, pages 205–210, 2002.

Ishikawa, Y., Subramanya, R. e Faloutsos, C. Mindreader: querying databases through multiple examples. In *VLDB Conference*, pages 218–227, 1998.

Jing, F., Li, M., Zhang, H. J. e Zhang, B. An efficient and effective region-based image retrieval framework. *Image Processing, IEEE Transactions on*, 13(5):699–709, 2004. ISSN 1057-7149.

Joachims, T. e Radlinski, F. Search engines that learn from implicit feedback. *Computer*, 40:34–40, 2007. ISSN 0018-9162.

Kay, K. N., Naselaris, T., Prenger, R. J. e Gallant, J. L. Identifying natural images from human brain activity. *Nature*, 452(7185):352–5, 2008.

Kherfi, M. L., Brahmi, D. e Ziou, D. Combining visual features with semantics for a more effective image retrieval. In *Pattern Recognition*, pages 961–964, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2128-2.

Kim, S., Park, S. e Kim, M. Central object extraction for object-based image retrieval. In *Proceedings of the 2nd international conference on Image and Video Retrieval*, CIVR'03, pages 39–49, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40634-4.

King, I. e Jin, Z. Integrated probability function and its application to content-based image retrieval by relevance feedback. *Pattern Recognition*, 36(9):2177–2186, 2003. ISSN 0031-3203.

Kosslyn, S. M. *Image and mind*. Harvard University Press, New York, 1980.

Kosslyn, S. M. e Thompson, W. L. *The Case for Mental Imagery*. Oxford Psychological Series. Oxford University Press, New York, 2006.

Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, 1 edition, 1992. ISBN 0262111705.

Laaksonen, J., Koskela, M. e Oja, E. Picsom-self-organizing image retrieval with mpeg-7 content descriptors. *IEEE Transactions on Neural Networks*, 13(4):841–853, 2002.

Leibe, B. e Schiele, B. Analyzing appearance and contour based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–415, 2003.

Lejsek, H., Ásmundsson, F., Jónsson, B. e Amsaleg, L. An efficient disk-based index for approximative search in very large high-dimensional collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):869–883, 2008.

Li, Y., Shapiro, L. O. e Bilmes, J. A. A generative/discriminative learning algorithm for image classification. In *IEEE International Conference on Computer Vision*, volume 2, pages 1605–1612, 2005.

Liu, D., Hua, K. A., Vu, K. e Yu, N. Fast query point movement techniques for large CBIR systems. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):729–743, 2009. ISSN 1041-4347. doi: http://dx.doi.org/10.1109/TKDE.2008.188.

Liu, Y., Zhang, D., Lu, G. e Ma, W. Y. Region-based image retrieval with perceptual colors. In *Pacific-Rim Multimedia Conference*, pages 931–938, 2004.

Liu, Y., Zhang, D., Lu, G. e Ma, W. Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40:262–282, 2007. ISSN 0031-3203.

Lowe, D. G. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

Lu, T. C. e Chang, C. C. Color image retrieval technique based on color features and image bitmap. *Information Processing and Management*, 43(2):461–472, 2007. ISSN 0306-4573. Special issue on AIRS2005: Information Retrieval Research in Asia.

Lu, Y. e Guo, H. Background removal in image indexing and retrieval. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, ICIAP '99, pages 933–938, Washington, DC, USA, 1999. IEEE Computer Society.

Luo, J. e Savakis, A. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. *International Conference on Image Processing (ICIP)*, II:745–748, 2001.

Ma, W. Y. e Manjunath, B. S. Netra: a toolbox for navigating large image databases. In *International Conference on Image Processing*, volume 1, pages 568–571, 1997.

Ma, W. Y. e Manjunath, B. S. Netra: a toolbox for navigating large image databases. In *Multimedia Systems*, volume 1, pages 568–571, 1999.

Marr, D. *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982.

Mehrotra, S., Rui, Y., Ortega-Binderberger, M. e Huang, T. S. Supporting content-based queries over images in mars. In *IEEE International Conference on Multimedia Computing and Systems*, pages 632–633, 1997.

Mezaris, V., Kompatsiaris, I. e Strintzis, M. G. An ontology approach to object-based image retrieval. In *IEEE International Conference on Image Processing*, pages 511–514, 2003.

Microsoft Research Cambridge. Object recognition image database 1.0. http://research.microsoft.com/vision/cambridge/recognition/.

Min, R. e Cheng, H. D. Effective image retrieval using dominant color descriptor and fuzzy support vector machine. *Pattern Recognition*, 42(1):147–157, 2009. ISSN 0031-3203.

Miranda, P. A. V., Torres, R. S. e Falcão, A. X. Tsd: A shape descriptor based on a distribution of tensor scale local orientation. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:139–146, 2005. ISSN 1530-1834.

Montoya-Zegarra, J. A., Leite, N. J. e Torres, R. S. Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–128, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2996-8.

Nene, S. A., Nayar, S. K. e Murase, H. Columbia university image library (coil-100). http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php.

Ogle, V. E. e Stonebraker, M. Chabot: retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, 1995. ISSN 0018-9162.

Palmeri, T. J. e Gauthier, I. Visual object understanding. *Nature Reviews Neuroscience*, 5:291–304, 2004.

Papa, J. P., Falcão, A. X. e Suzuki, C. T. N. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009. ISSN 0899-9457.

Papa, J. P., Cappabianco, F. A. M. e Falcão, A. X. Optimizing optimum-path forest classification for huge datasets. In *International Conference on Pattern Recognition*, pages 4162–4165, Washington, DC, USA, 2010.

Penatti, O. A. B. Estudo comparativo de descritores para recuperação de imagens por conteudo na web. Dissertação de mestrado, Unicamp, 2009.

Pentland, A., Picard, R. W. e Sclaroff, S. Photobook: content-based manipulation of image databases. *International Journal of Computer Vision*, 18:233–254, 1996. ISSN 0920-5691.

Philipp-Foliguet, S., Gony, J. e Gosselin, P. H. FReBIR: an image retrieval system based on fuzzy region matching. *Computer Vision and Image Understanding*, 113(6):693–707, 2009.

Porkaew, K., Chakrabarti, K. e Mehrotra, S. Query refinement for multimedia similarity retrieval in mars. In *Proceedings of ACM Multimedia*, pages 235–238, 1999.

Pylyshyn, Z. Is the imagery debate over? what was it about?, 2000.

Ragnemalm, I. The euclidean distance transform in arbitrary dimensions. *Pattern Recognition Letters*, 14(11):883–888, 1993.

Ranganath, C. Working memory for visual objects: Complementary roles of inferior temporal, medial temporal, and prefrontal cortex. *Neuroscience*, 139(1):277–289, 2006.

Rocchio, J. J. *Relevance feedback in information retrieval*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

Rui, Y. e Huang, T. Optimizing learning in image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 236–243, 2000.

Rui, Y., Huang, T. S. e Mehrotra, S. Content-based image retrieval with relevance feedback in mars. In *IEEE International Conference on Image Processing*, pages 815–818, 1997.

Rui, Y., Huang, T. S., Ortega, M. e Mehrotra, S. Relevance feedback: A power tool for interactive content-based image retrieval. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 8 (5), pages 644–655, 1998.

Rui, Y., Huang, T. S. e Chang, S. F. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, (10):39–62, 1999.

Ruppert, G. C. S., Favretto, F. O., Falcão, A. X., Yassuda, C. L. e Bergo, F. P. G. Fast and accurate image registration using the multiscale parametric space and grayscale watershed transform. In *Systems, Signals and Image Processing*, pages 17–19, Rio de Janeiro, June 2010. IEEE Computer Society.

Serre, T. e Poggio, T. A neuromorphic approach to computer vision. *Communications of the ACM*, 53:54–61, October 2010. ISSN 0001-0782.

Shastri, L. A computational model of episodic memory formation in the hippocampal system. *Neurocomputing*, 38:38–40, 2001.

Shi, J. e Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000. ISSN 0162-8828.

Sikora, T. The mpeg-7 visual standard for content description-an overview. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):696–702, 2001. ISSN 1051-8215.

Silva, A. T., Falcão, A. X. e Magalhães, L. P. A new CBIR approach based on relevance feedback and optimum-path forest classification. *Journal of WSCG*, 18(1-3):73–80, feb 2010. ISSN 1213-6972.

Silva, A. T., Falcão, A. X. e Magalhães, L. P. Active learning paradigms for CBIR systems based on optimum-path forest classification. *Pattern Recognition*, 44(12):2971–2978, 2011. ISSN 0031-3203.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A. e Jain, R. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22: 1349–1380, 2000.

Smith, J. R. e Chang, S. F. VisualSEEk: a fully automated content-based image query system. In *ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 87–98, New York, NY, USA, 1996. ACM. ISBN 0-89791-871-1.

Snoek, C. G. M. e Smeulders, A. W. M. Visual-concept search solved? *IEEE Computer*, 43(6): 76–78, 2010.

Srinivasa, K. G., Sridharan, K., Shenoy, P. D., Venugopal, K. R. e Patnaik, L. M. A neural network based CBIR system using sti features and relevance feedback. *Intelligence Data Analysis*, 10: 121–137, March 2006.

Stehling, R. O., Nascimento, M. A. e Falcão, A. X. A compact and efficient image retrieval approach based on border/interior pixel classification. In *International Conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM. ISBN 1-58113-492-4.

Su, J. H., Huang, W. J., Yu, P. S. e Tseng, V. S. Efficient relevance feedback for content-based image retrieval by mining user navigation patterns. *IEEE Trans. on Knowl. and Data Eng.*, 23:360–372, March 2011. ISSN 1041-4347.

Takala, V., Ahonen, T. e Pietikäinen, M. Block-based methods for image retrieval using local binary patterns. In *Scandinavian Conference on Image Analysis (SCIA)*, pages 882–891, 2005.

Tao, B. e Dickinson, B. W. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327–342, 2000. ISSN 1047-3203.

Thomas, N. J. T. Mental imagery. In Zalta, Edward N., editor, *The Stanford Encyclopedia of Philosophy*. Fall 2010 edition, 2010.

Titterington, D. M., Smith, A. F. M. e Makov, U. E. *Statistical Analysis of Finite Mixture Distributions*. John Wiley, New York, 1985.

Tong, S. e Chang, E. Support vector machine active learning for image retrieval. In *ACM International Conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM. ISBN 1-58113-394-4.

Torres, R. S. e Falcão, A. X. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

Torres, R. S., Falcão, A. X. e Costa, L. F. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004. ISSN 0031-3203.

Torres, R. S., Falcão, A. X., Gonçalves, M. A., Papa, J. P., Zhang, B., Fan, W. e Fox, E. A. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, Feb 2009.

Town, C. e Sinclair, D. Content based image retrieval using semantic visual categories. Technical report, Society for Manufacturing Engineers, 2001.

Traina Jr., C., Traina, A., Faloutsos, C. e Seeger, B. Fast indexing and visualization of metric data sets using slim-trees. *Knowledge and Data Engineering, IEEE Transactions on*, 14(2):244 –260, 2002. ISSN 1041-4347.

Tuytelaars, T. e Mikolajczyk, K. Local invariant feature detectors: a survey. *Foundation and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008. ISSN 1572-2740.

Vadivel, A., Majumdar, A. e Sural, S. Characteristics of weighted feature vector in content-based image retrieval applications. In *Intelligent Sensing and Information Processing*, number 18 in 1, pages 127–132, 2004.

Valle, E. e Cord, M. Advanced techniques in CBIR local descriptors, visual dictionaries and bag of features. In *Tutorials of 22nd Sibgrapi*, 2009.

Valle, E., Cord, M. e Philipp-Foliguet, S. High-dimensional descriptor indexing for large multimedia databases. In *ACM Conference on Information and Knowledge Management*, pages 739–748, New York, NY, USA, 2008. ACM.

Vasconcelos, N. Content-based retrieval from image databases: current solutions and future directions. In *International Conference in Image Processing (ICIP)*, pages 6–9, 2001.

Veltz, F. Image mining: accelerated visual analysis. In *Défense Nationale*, pages 101–112, 2004.

Vieira, M. R., Traina Jr, C., Chino, F. J. T. e Traina, A. J. M. DBM-Tree: A dynamic metric access method sensitive to local density data. *Journal of Information and Data Management*, 1(1):111–128, 2010.

Wang, H. H., Mohamad, D. e Ismail, N. A. Semantic gap in CBIR: Automatic objects spatial relationships semantic extraction and representation. *International Journal of Image Processing*, 4: 192–204, 2010.

Wang, J. Z., Wiederhold, G., Firschein, O. e Wei, S. X. Content-based image indexing and searching using daubechies' wavelets. *International Journal on Digital Libraries*, 1:311–328, 1998. ISSN 1432-5012.

Wang, J. Z., Li, J. e Wiederhold, G. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

Wang, X. Y., Chen, J. W. e Yang, H. Y. A new integrated SVM classifiers for relevance feedback content-based image retrieval using EM parameter estimation. *Applied Soft Computing*, 11:2787–2804, March 2011. ISSN 1568-4946.

Williams, A. e Yoon, P. Content-based image retrieval using joint correlograms. *Multimedia Tools and Applications*, 34(2):239–248, 2007. ISSN 1380-7501.

Wu, J., Lin, Z. K. e Lu, M. Y. Asymmetric semi-supervised boosting for SVM active learning in CBIR. In *ACM International Conference on Image and Video Retrieval*, CIVR '10, pages 182–188, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0117-6.

Xu, Z. e Akella, R. Active relevance feedback for difficult queries. In *ACM Conference on Information and Knowledge Management*, CIKM '08, pages 459–468, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3.

Zhang, D. e Lu, G. A comparative study on shape retrieval using fourier descriptors with different shape signatures. *Journal of Visual Communication and Image Representation*, 1(14):41–60, 2003.

Zhang, L., Liu, F. e Zhang, B. Support vector machine learning for image retrieval. *International Conference on Image Processing*, pages 7–10, 2001.

Zhou, Z. H., Chen, K. J. e Dai, H. B. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Transactions on Information Systems*, 24:219–244, 2006.

Apêndice A

Trabalhos aceitos e submetidos até a data da defesa

A seguir são apresentados os artigos publicados e submetidos pelo autor da tese até a data da defesa. Estes trabalhos são resultados do desenvolvimento realizado ao longo desta tese.

1. A.T. Silva, A.X. Falcão, L.P. Magalhães. "A new CBIR approach based on relevance feedback and optimum-path forest classification". *Journal of WSCG*, 18 (1-3), pg. 73–80, 2010.

2. A.T. Silva, A.X. Falcão, L.P. Magalhães. "Active learning paradigms for CBIR systems based on optimum-path forest classification". *Pattern Recognition*, 44 (12) pg. 2971–2978, 2011.

3. J. A. dos Santos, A. T. da Silva, R. da S. Torres, A. X. Falcão, L. P. Magalhães, R. A. C. Lamparellic. "Interactive Classification of Remote Sensing Images by using Optimum-Path Forest and Genetic Programming". *Poster of the International Conference on Computer Analysis of Images and Patterns*.

4. A.T. Silva, J. A. dos Santos, A.X. Falcão, R. da S. Torres, L.P. Magalhães. "Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning". *Computer Vision and Image Understanding* (submetido em dezembro de 2010).

A new CBIR approach based on relevance feedback and optimum-path forest classification

André Tavares da Silva
Unicamp, Brazil
atavares@dca.fee.unicamp.br

Alexandre Xavier Falcão
Unicamp, Brazil
afalcao@ic.unicamp.br

Léo Pini Magalhães
Unicamp, Brazil
leopini@dca.fee.unicamp.br

ABSTRACT

Recently some CBIR approaches have shown the use of relevance feedback to train a pattern classifier to select relevant images for retrieval. This paper revisits this strategy by using an optimum-path forest (OPF) classifier. During relevance feedback iterations, the proposed method uses the OPF classifier to decide which database images are relevant or not. Images classified as relevant are sorted and presented to the user for a new iteration. Such images are ordered according to the normalized distance using relevant and irrelevant representative images, computed previously by the OPF classifier. Our experiments show that the proposed approach requires fewer iterations, being faster and more effective than methods based on SVM.

Keywords: CBIR, Relevance Feedback, OPF.

1 INTRODUCTION

Image collections have been created and used in several applications, such as digital libraries, medicine, and biodiversity information systems. Given the size of these collections, it is essential to provide efficient and effective means to retrieve images. Such a problem has raised the interest in putting together image processing, information retrieval, and database management to design content-based image retrieval (CBIR) systems for large image collections [2].

The visual content of an image in a CBIR system is often represented by a *feature vector*, which may encode color, texture, and/or shape measures. The image is then interpreted as a *point* in the feature space. A query in a CBIR system is usually done by range (returning all images whose distance to the query image is less than a given radius) or by similarity. We will focus on the second approach where a number of the closest images to the query point are retrieved from the database. Given that the meaning of those images may differ for distinct users since they are not completely represented by the feature vector, a *semantic gap* occurs between the user's expectation and the result of the query. Thus, *relevance feedback* techniques have been investigated to reduce the semantic gap by requiring more user interaction than simply the specification of a query image. These techniques usually involve three steps: (i) a small number of retrieved images is presented to the user; (ii) the user indicates which images are relevant; (iii) the system learns the user's opinion from this feedback in order to return more relevant images in the next iteration. This process may be repeated until the user is satisfied, but it is highly desirable to finish it in a few iterations.

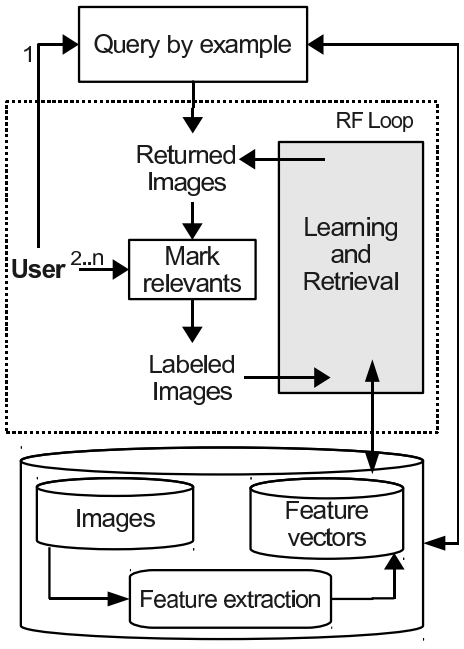


Figure 1: CBIR with Relevance Feedback.

Figure 1 shows an overview of the relevance feedback process. There are several studies on each stage of this pipeline, such as creating more robust local descriptors[15, 19] (i.e. feature vectors and distance functions to compare them) or providing scalability to

huge image databases[9, 20]. Our work focuses on the learning and retrieval process (gray box in Figure 1), especially in query classification and ranking.

Relevance feedback techniques were initially proposed for document retrieval, but have been successfully applied to CBIR systems[2, 13, 14, 17, 23]. Figure 2 illustrates three examples of simple relevance feedback techniques [7, 10]. In Figure 2a, the positive examples (relevant images) from a first iteration are used to move the next query point to their geometric center in the feature space. This idea stemmed from Rocchio's formula [13] in document retrieval and it has been successfully exploited in CBIR systems, such as MARS [14] and MindReader [6]. Two other methods use the relevant images as next query points and, depending on the distance to this multi-point query set, different isosurfaces are formed in the feature space (Figures 2b and 2c). The method we present here is also a multi-point query but, differently from those approaches, we exploit relevant and irrelevant images as query points.

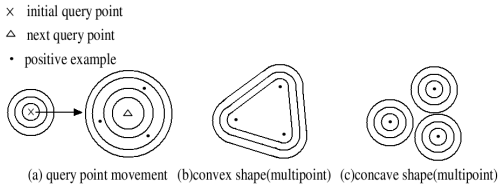


Figure 2: Simple relevance feedback techniques that change query shapes (i.e., isosurfaces with respect to the query points).

Approaches based on relevant and irrelevant images usually exploit active learning techniques to design a classifier that selects from the database the candidate relevant images for sorting by distance to the query point [1, 3, 16]. The method proposed by Tong and Chang. [16] uses Support Vector Machines (SVM) for image classification [22]. During the relevance feedback iterations, the method finds the optimum hyperplane that separates relevant and irrelevant images, presenting to the user the images closer to the hyperplane. This hyperplane is adjusted along the iterations and, after a last iteration, the method presents the images farther to the hyperplane, on its relevant side.

The method we propose here follows a similar strategy, using a faster and more effective classifier aiming to present the most relevant images in the database at each iteration, unlike SVM. For a given set of relevant and irrelevant images, the method designs an OPF (Optimum-Path Forest) classifier [12]. Only database images classified as relevant are sorted by distance and presented to the user in the next iteration. This distance is computed based on relevant and irrelevant prototypes (representative images), computed during the training of OPF classifier. We show that this strategy is actually very effective reducing considerably the number of required iterations.

In addition to that, the *distance function* used to compare images has also influence on the retrieval process. Some methods use multiple pairs of feature vectors and distance functions, called *descriptors*, and compare two images by combining their distance based on each descriptor [11, 18]. In this case, the learning from relevance feedback may also change the way to combine descriptors [18]. Our method can exploit the same framework, but we will consider in this study only a single descriptor per image.

This paper is organized as follows. Section 2 presents the proposed algorithm based on OPF classifier and an example illustrates our relevance feedback process. The experiments and results using three heterogeneous image databases are described in Section 3. As baselines for comparison, we use the method od Tong and Chang [16] and the one illustrated in Figure 2c, which uses only relevant images for multi-point query. Section 4 states the conclusions and discusses our future work.

2 CBIR USING OPF CLASSIFIER

OPF is a classification method which represents each class of objects by one or more optimum-path trees rooted at given samples, called prototypes [12]. The training samples are nodes of a complete graph, whose arcs are weighted by the distance between the feature vectors of their nodes. In relevance feedback, we have two classes: relevant images chosen by the user and irrelevant ones. The prototypes computed by the OPF classifier are then used to sort the images according to the user's selection.

Let $\mathscr{Z}$ be an image database. For every image $t \in \mathscr{Z}$, we have a feature vector $\vec{v}(t) \in \Re^n$. That is, every image may be interpreted as a point in the feature space $\Re^n$. The distance $d(s,t)$ between two images s and t is the distance between their corresponding feature vectors. For an initial query point s, the proposed method returns the N closest images in $\mathscr{Z}$ to s (query by similarity). Due to the semantic gap, the closest images to s may not be the most relevant for a given user. By marking the relevant images among the returned ones, the user creates two sets: a set $\mathscr{I} \subset \mathscr{Z}$ of irrelevant images and a set $\mathscr{R} \subset \mathscr{Z}$ of relevant images. The method then uses sets $\mathscr{R}$ and $\mathscr{I}$ to compute two optimum-path forests (OPF), one for each class. Each database image $t \in \mathscr{Z} \backslash \mathscr{I} \cup \mathscr{R}$ is then classified according to the root's label of the forest (relevant/irrelevant) which offers to t the optimum path in the graph. Only the N closest images labeled as relevant will be returned in a set $\mathscr{C}$ to the user in the next iteration. Relevant prototypes ($\mathscr{A}$) and irrelevant ones ($\mathscr{B}$), computed in the previous step, are then used to sort the images in $\mathscr{C}$ for the next iteration.

The method computes the average distance $\bar{d}_{\mathscr{A}}(t,\mathscr{A})$ between each image $t \in \mathscr{C}$ and images in the set of relevant prototypes $\mathscr{A}$. It also computes the average distance $\bar{d}_{\mathscr{B}}(t,\mathscr{B})$ between t and images in the set of irrelevant prototypes $\mathscr{B}$. Finally, a distance $\bar{d}(t,\mathscr{A},\mathscr{B})$ is computed as a normalized mean between relevant and irrelevant prototypes: $\bar{d}(t,\mathscr{A},\mathscr{B}) = \frac{\bar{d}_{\mathscr{A}}(t,\mathscr{A})}{\bar{d}_{\mathscr{A}}(t,\mathscr{A})+\bar{d}_{\mathscr{B}}(t,\mathscr{B})}$.

Algorithm 1: Relevance Feedback Algorithm

Input: A query image s, a feature extraction function v, a distance function d, a desirable number N of relevant images, an image database $\mathscr{Z}$ and a number T of iterations.
Output: An ordered list L of the N most relevant images in $\mathscr{Z}$.
Auxiliary: Sets $\mathscr{R} \subset \mathscr{Z}$ and $\mathscr{I} \subset \mathscr{Z}$ of relevant and irrelevant images, $\mathscr{A} \subset \mathscr{Z}$ and $\mathscr{B} \subset \mathscr{Z}$ of relevant and irrelevant prototypes, set $\mathscr{C} \subset \mathscr{Z}$ of images classified as relevant for the next iteraction.

1 Compute the distance $d(s,t)$ for every image $t \in \mathscr{Z}$.
2 Create an ordered list L of the N closest images t to s based on $d(s,t)$.
3 Set $\mathscr{I} \leftarrow \emptyset$ and $\mathscr{R} \leftarrow \emptyset$.
4 **for** *each learning iteration* $i = 1,2,\ldots,T$ **do**
5 Set $\mathscr{C} \leftarrow \emptyset$.
6 The user marks the relevant images in L, which are inserted into $\mathscr{R}$ and the irrelevant ones are inserted into $\mathscr{I}$.
7 **if** $|\mathscr{R}| < N$ **then**
8 Compute OPF using sets $\mathscr{I}$ and $\mathscr{R}$, resulting also $\mathscr{A}$ and $\mathscr{B}$.
9 **for** *each image* $t \in \mathscr{Z} \setminus \mathscr{I} \cup \mathscr{R}$ **do**
10 **if** t *is labeled as relevant by OPF* **then**
11 insert t into the set $\mathscr{C}$ of images classified as relevant.
12 **end**
13 **end**
14 **end**
15 **else**
16 Return the final ordered list L with the N most relevant images in $\mathscr{R}$, as defined by the user's selection.
17 **end**
18 Create an ordered list L with the N most relevant images in $\mathscr{C}$, in increasing order of $\bar{d}(t,\mathscr{A},\mathscr{B})$.
19 **end**
20 Return the final ordered list L with the N most relevant images in $\mathscr{R}$, completing it with the $N - |\mathscr{R}|$ relevant images in $\mathscr{C}$ in the increasing order of $\bar{d}(t,\mathscr{A},\mathscr{B})$.

After classifying each image in $\mathscr{Z} \setminus \mathscr{I} \cup \mathscr{R}$, the method returns to the user a new set of N relevant images, which contains the lowest values of $\bar{d}(t,\mathscr{A},\mathscr{B})$. This process is then repeated for a few iterations T and, finally, the system returns all relevant images obtained so far.

In order to illustrate the advantages of our relevance feedback approach as compared to a simple retrieval of the N closest images to s, we present an example of query image in Figure 3 from the image database Corel [21]. We use a color descriptor, called BIC, proposed by Stehling et al. [15]. The $N = 30$ closest images in that database are shown in Figure 16, where the relevant images are presented with a blue border. After $T = 3$ iterations (a reasonable number of iterations for practical situations), the system presents the $N = 30$ most relevant images found so far, as shown in Figure 17. It is important to note that the quality of this result may vary depending on the image descriptor.

Figure 3: A query image s.

3 EXPERIMENTS AND RESULTS

In order to evaluate our method, we use the BIC descriptor with the dLog distance function [15], and compare its effectiveness using precision-recall curves and two other approaches as baselines: the SVM-based method proposed by Tong and Chang [16] and the multi-point query with relevant images only, as illustrated in Figure 2c. The first, named here as SAL (SVM Active Learning), is also named SVM_{ACTIVE} or SVM_{AL} in the literature. It was chosen because it is based on a state-of-the-art technique for image classification. The second, named as QPM (Query Point Movement) [16], was selected to illustrate the importance of irrelevant images in the multi-point query set. Our approach is named here OPF_{AL} or simply OPF, because it is based on the OPF classifier.

As mentioned before, our work focuses in query classification and ranking. Indexing schemes to accelerate the search can be exploited in our method, as well as techiques for descriptor combination. But we consider in this study only a single descriptor in order to compare the proposed method against others.

The curves of precision-recall use the entire image database $\mathscr{Z}$. Thus, lines 18 and 20 of our algorithm are replaced by: Create a list L with all relevant images in $\mathscr{C} \cup \mathscr{R}$, in their increasing order of $\bar{d}(t, \mathscr{A}, \mathscr{B})$, and compute the precision-recall curve for images in L.

The experiments used three heterogeneous image databases, representing different challenges for CBIR.

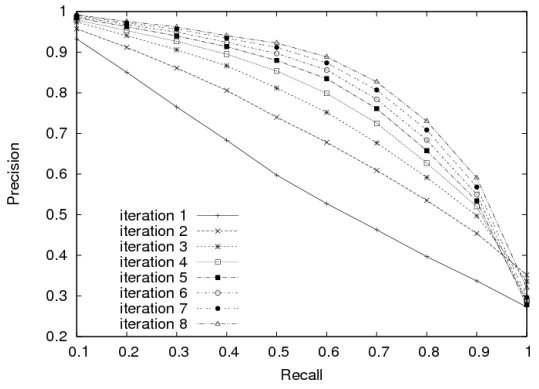


Figure 4: Mean precision-recall curves of OPF in Corel database, iterations 1 to 8.

- PASCAL [4, 5].

  This database consists of 3,448 natural images, each one containing multiple regions of interest (subimages). Each region contains one object from a class of visual objects (bikes, boats, birds). The regions are labeled by their class performing a total of 23 classes with different number of images, varying from 72 to 446 subimages each.

- Corel [21].

  This database is a collection with 200,000 images from the Corel GALLERY Magic–Stock Photo Library 2. We use a subset of 3,906 natural images, pre-classified into 85 classes. These classes have different number of images varying from 7 to 98 images each.

- ETH-80 [8].

  This database is available in the project COGVIS, serving for both psychophysical and computational studies concerning object recognition and categorization. The project includes images of objects from 8 basic-level categories performing a total of 2,384 images, distributed uniformly among the classes.

For each image database, we simulate the user behavior by using each image as initial query point and marking the relevant points (images from the same class of the query) from 30 returned images at each iteration.

First, we present in Figures 4, 5 and 6 the mean precision-recall curves of OPF for the databases Corel, ETH-80, and PASCAL, respectively, by varying the number of iterations from 1 to 8. These curves show that OPF improves its performance (the higher the precision-recall curve, the better is the method) with the number of iterations, as expected, but they also indicate the challenge degree of each database: PASCAL imposes more challenges than Corel which is more difficult than ETH-80.

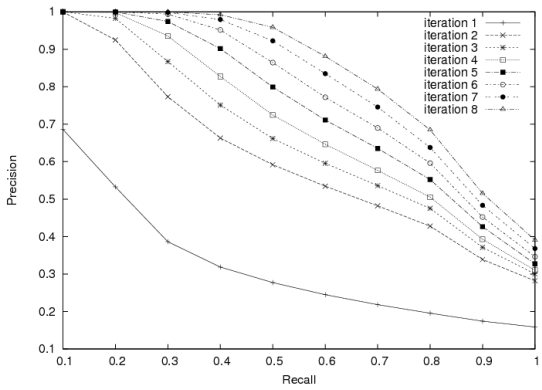


Figure 5: Mean precision-recall curves of OPF in ETH-80 database, iterations 1 to 8.

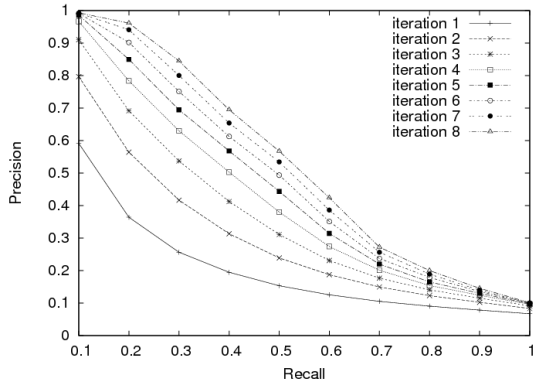


Figure 6: Mean precision-recall curves of OPF in PASCAL database, iterations 1 to 8.

Following, Figures 7 to 15 show the mean precision-recall curves of each method (OPF, QPM, and SAL) in each database (Corel, ETH-80, and PASCAL) for 3, 5 and 8 relevance feedback iterations. One may observe that OPF outperformed SAL and QPM in the most difficult databases, Corel and Pascal, and for all number of iterations. In the easiest case, ETH-80, the curves cross each other in some recall rates, but OPF is still better than the others up to 40% of recall for 3 and 5 iterations, and 50% of recall for 8 iterations. In addition to that, OPF is much faster than SAL and it learns quicker the simulated user's wish, providing effective results in fewer iterations. We consider 3 iterations as the ideal

number for practical situations. OPF has also outperformed QPM in all cases and this indicates the importance of using relevant and irrelevant points in multi-point query systems rather than only relevant points.

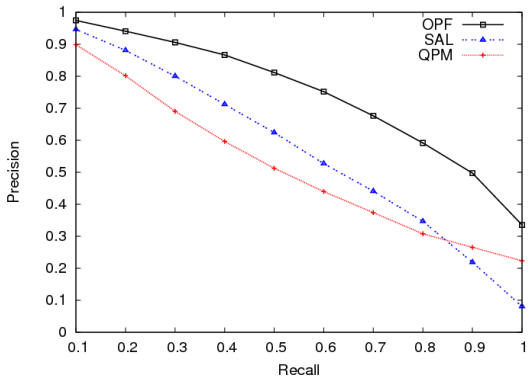


Figure 7: Mean precision-recall curves in Corel database, third iteration.

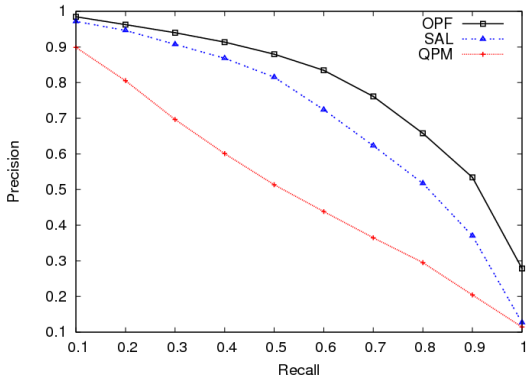


Figure 8: Mean precision-recall curves in Corel database, fifth iteration.

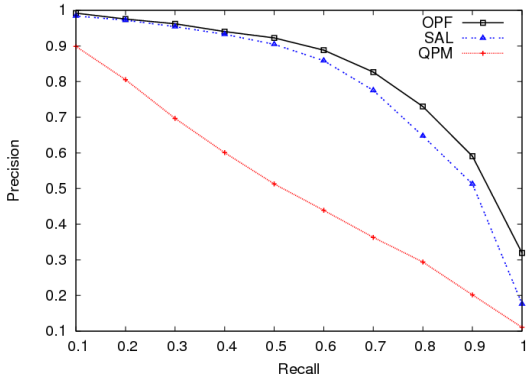


Figure 9: Mean precision-recall curves in Corel database, eighth iteration.

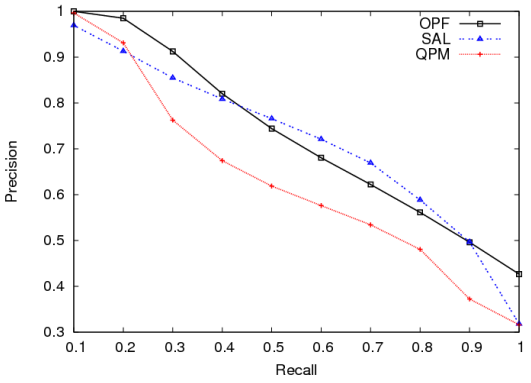


Figure 10: Mean precision-recall curves in ETH-80 database, third iteration.

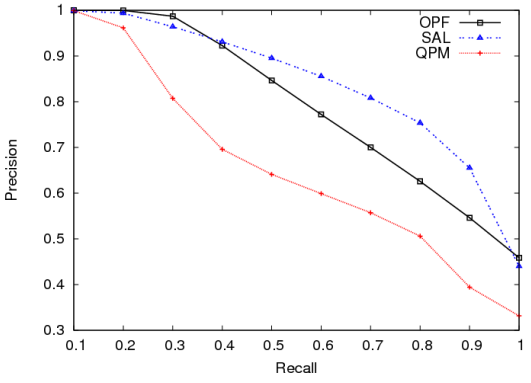


Figure 11: Mean precision-recall curves in ETH-80 database, fifth iteration.

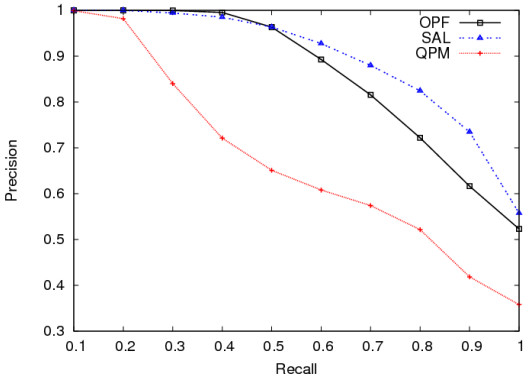


Figure 12: Mean precision-recall curves in ETH-80 database, eighth iteration.

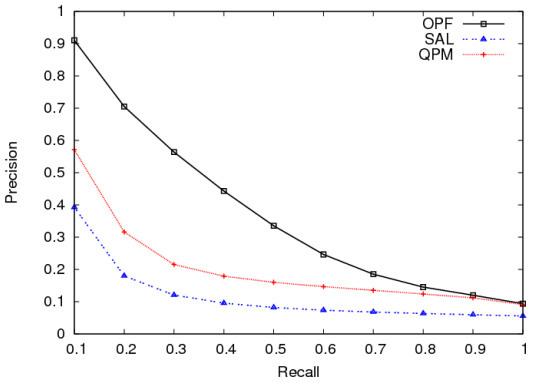


Figure 13: Mean precision-recall curves in PASCAL database, third iteration.

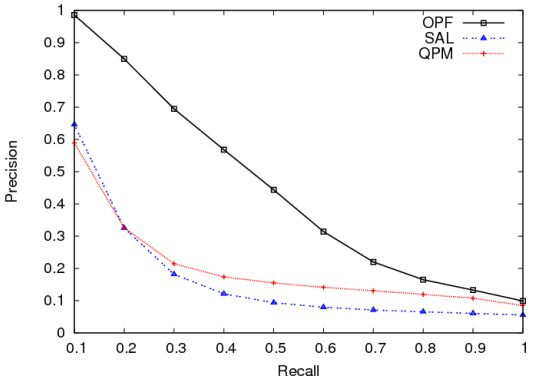


Figure 14: Mean precision-recall curves in PASCAL database, fifth iteration.

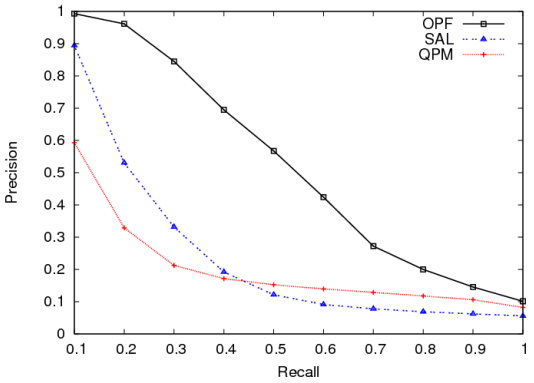


Figure 15: Mean precision-recall curves in PASCAL database, eighth iteration.

Tables 1 and 2 show the execution times of SAL and OPF, respectively. We present time values for all images of the databases for iterations 3, 5 and 8, used to compute the precision-recall curves of Figures 7 to 15.

Table 3 shows the average execution time for one iteration of SAL and OPF. Our approach could also be used with indexing schemes to further accelerate the search process and reduce the execution time. However, we present in this paper execution times without indexing structures. As the number of iterations grows, the runtime increases. In the Corel database for instance, our method takes 0.4 seconds to present images at the eighth iteration while SAL takes 10.2 seconds in the average. The tests were performed in a machine with Intel Pentium D processor at 3.4GHz and 1 GB RAM running the Linux operational system.

Table 1: Total execution time of SAL (minutes).

Database	Corel	ETH-80	PASCAL
3 iterations	1,116	420	903
5 iterations	2,170	702	1,743
8 iterations	5,330	1,120	4,483

Table 2: Total execution time of OPF (minutes).

Database	Corel	ETH-80	PASCAL
3 iterations	42.8	24.2	33.4
5 iterations	102.9	38.9	81.2
8 iterations	224.0	59.1	188.4

Table 3: Average execution time per query (seconds).

Database	Corel	ETH-80	PASCAL
SAL	5.71	3.53	1.96
OPF	0.22	0.20	0.19

Methods such as OPF and SAL classify candidate relevant images in the image database and sort them to select the N closest to the query point(s). One may ask about the relevant images misclassified as irrelevant. These images are lost by the system. Table 4 presents the percentage of images that were erroneously discarded by OPF for each database and for iterations 3, 5 and 8; the percentages of missed relevant images are insignificant. This result is even more important when we consider that the performance of CBIR systems, as mentioned before, usually increases with the number of descriptors and strategies to combine them [18], which is not being exploited in the present study.

Table 4: Percentages of relevant images missed by OPF due to classification in each database.

Database	Corel	ETH-80	PASCAL
3 iterations	0.36%	1.23%	3.37%
5 iterations	0.32%	1.21%	3.22%
8 iterations	0.27%	1.02%	3.06%

4 CONCLUSION AND FUTURE WORK

We presented a new relevance feedback technique for CBIR. This is the first time that the OPF classifier is being used and evaluated for small training sets, as required in learning by relevance feedback. Differently from the original method, we have separated the optimum-path forests of each class for classification. This constitutes a simple but very effective variant of the original method. We have also proposed a new order relation among the relevant images, which is based on the mean distances to the prototypes of the OPF classifier.

We have evaluated the method using a color descriptor, three heterogeneous image databases, two reference approaches, a few iterations, and query by similarity. The results indicated that the proposed method, named OPF, requires fewer iterations of relevance feedback. It outperformed the reference approaches in all databases and the number of missed relevant images due to classification was insignificant.

The new CBIR approach based on relevance feedback and optimum-path forest classification presented in this paper provides a solution in interactive time for practical applications. On average our method was twenty times faster than SAL.

Our future work involves the use of multiple descriptors and techniques to combine them. We intend to use other descriptors based on shape, texture and color and combine them by using techniques such as Bayesian framework, Genetic Programming [18] or other similar approaches. We also intend to investigate other image classifiers and to evaluate the methods for multiple users.

ACKNOWLEDGEMENTS

The first author thanks CNPq for financial support (140968/2007-5). The second author thanks CNPq (project ARPIs, 481556/2009-5) and FAPESP (07/52015-0).

REFERENCES

[1] M. Cord, J. Fournier, and S. Philipp-Foliguet. Exploration and search-by-similarity in cbir. In *SIBGRAPI '03: Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 175–182. IEEE Computer Society, 2003.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[3] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on bayesian inference for image retrieval. *Signal Process.*, 85(2):395–399, 2005.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[6] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice Hall, 2002.

[7] D.-H. Kim and C.-W. Chung. Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 599–610, 2003.

[8] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pages 409–415, 2003.

[9] H. Lejsek, F. Ásmundsson, B. Jónsson, and L. Amsaleg. An efficient disk-based index for approximative search in very large high-dimensional collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):869–883, 2008.

[10] D. Liu, K. A. Hua, K. Vu, and N. Yu. Fast query point movement techniques for large cbir systems. *IEEE Trans. on Knowl. and Data Eng.*, 21(5):729–743, 2009.

[11] R. Ohbuchi and Yushin Hata. Combining multiresolution shape descriptors for 3d model retrieval. In *Proc. WSCG 2006*, Plzen, Czech Republic, 2006.

[12] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.

[13] J. J. Rocchio. *Relevance feedback in information retrieval*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

[14] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in mars. In *In Proc. IEEE Int. Conf. on Image Proc*, pages 815–818, 1997.

[15] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM.

[16] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM.

[17] R. S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[18] R.S. Torres, A.X. Falcão, M.A. Gonçalves, J.P. Papa, B. Zhang, W. Fan, and E.A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):217–312, Feb 2009.

[19] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, 2008.

[20] E. Valle, M. Cord, and S. Philipp-Foliguet. High-dimensional descriptor indexing for large multimedia databases. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 739–748, New York, NY, USA, 2008. ACM.

[21] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[22] L. Zheng and X. He. Classification techniques in pattern recognition. In *Proc. WSCG 2005*, London, United Kingdom, 2005.

[23] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.

Figure 16: Closest images to s based on $d(s,t)$.

Figure 17: Result of OPF after three iterations.

ARTICLE IN PRESS

Pattern Recognition ▮ (▮▮▮▮) ▮▮–▮▮

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Active learning paradigms for CBIR systems based on optimum-path forest classification

André Tavares da Silva [a,*], Alexandre Xavier Falcão [b], Léo Pini Magalhães [a]

[a] Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, University of Campinas (Unicamp), 400 Albert Einstein Avenue, 13083-970 Campinas, SP, Brazil
[b] Institute of Computing, University of Campinas (Unicamp), 1251 Albert Einstein Avenue, 13083-852 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 25 October 2010
Received in revised form
7 February 2011
Accepted 26 April 2011

Keywords:
Content-based image retrieval
Relevance feedback
Optimum-path forest classifiers
Active learning
Image pattern analysis

ABSTRACT

This paper discusses methods for content-based image retrieval (CBIR) systems based on relevance feedback according to two active learning paradigms, named *greedy* and *planned*. In greedy methods, the system aims to return the most relevant images for a query at each iteration. In planned methods, the most informative images are returned during a few iterations and the most relevant ones are only presented afterward. In the past, we proposed a greedy approach based on optimum-path forest classification (OPF) and demonstrated its gain in effectiveness with respect to a planned method based on support-vector machines and another greedy approach based on multi-point query. In this work, we introduce a planned approach based on the OPF classifier and demonstrate its gain in effectiveness over all methods above using more image databases. In our tests, the most informative images are better obtained from images that are classified as relevant, which differs from the original definition. The results also indicate that both OPF-based methods require less user involvement (efficiency) to satisfy the user's expectation (effectiveness), and provide interactive response times.

1. Introduction

Content-based image retrieval (CBIR) systems aim to return the most relevant images in a database, according to the user's opinion for a given query. Due to the dynamic nature of the problem, which may change the meaning of relevance among users for a same query, these systems usually rely on an active learning process in which the system returns a small set of images (training set) and the user indicates their relevance at each iteration (see Fig. 1) [1]. The database images can be represented by *feature vectors* (points in a feature space), that may encode color, texture, and/or shape measures, using *indexing structures* [2,3] to access images in a more efficient way. There are many research activities on each stage shown in Fig. 1. Examples are works to obtain more effective image descriptors [4,5] (i.e., feature extraction and distance functions for image comparison), to combine distance functions from multiple descriptors [6,7], and to provide scalability in large image databases [8,9]. The methods presented here can take advantage of all these results, but the focus of our work is on the learning and retrieval processes (gray box in Fig. 1).

From the practical point of view, a CBIR system should minimize the response time and the number of marked images (efficiency), while it maximizes the user's satisfaction (effectiveness). These constitute the main challenges, especially when we consider large image collections. We have observed two active learning paradigms from relevance feedback on returned images, named *greedy* and *planned*. In greedy methods,

1. a small number of images, usually ranked by relevance (similarity with the query), is presented to the user,
2. the user indicates which images are actually relevant (irrelevant), being the complement understood as irrelevant (relevant) images,
3. the system learns the user's opinion from this feedback, in order to return a higher number of relevant images in a next iteration at step 1.

In planned methods, the user establishes in which iteration the system should return images ranked by relevance. In the previous iterations, the system presents the *most informative* images in order to better learn the distribution of the relevant and irrelevant classes in the feature space (i.e., to train a pattern classifier).

In the past, we proposed a greedy approach [10] based on the optimum-path forest classifier (OPF) [11]. In this method, database images classified as relevant are ranked based on their *normalized distances* to special positive and negative examples (called *prototypes*), which are computed in the previous iteration

* Corresponding author.
E-mail addresses: atavares@dca.fee.unicamp.br (A.T. da Silva), afalcao@ic.unicamp.br (A.X. Falcão), leopini@fee.unicamp.br (L.P. Magalhães).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.026

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

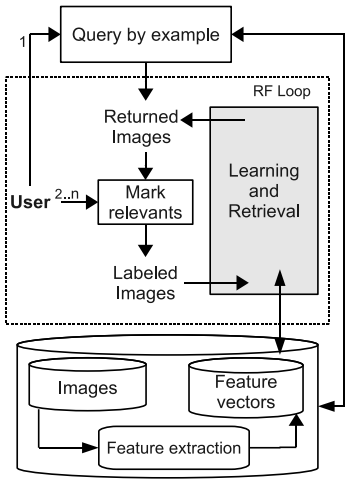


Fig. 1. A CBIR system with relevance feedback [10].

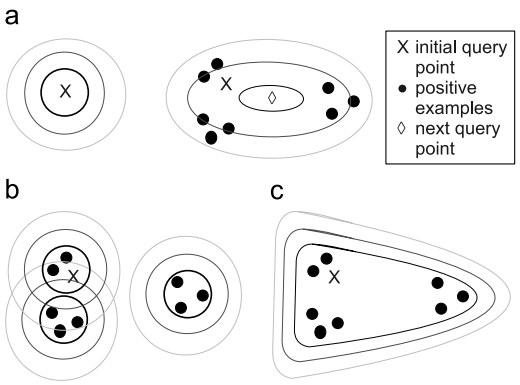


Fig. 2. Simple greedy techniques that change query shapes (i.e., iso-surfaces with respect to the query points). (a) First iteration and query point movement. (b) Convex shape (multipoint). (c) Concave shape (multipoint).

from the user-marked images. Its effectiveness gain was notorious over a simple greedy technique [12] and a planned method based on support-vector machine (SVM) [13]. The choice of the OPF model was also justified by its considerable gain in computational time with respect to SVM and other classification models, such as neural networks [11].

In this work, we present a planned method based on the OPF classifier and demonstrate its gain in effectiveness over the previous approaches [10,13,12] using more image databases. In our tests, the most informative images are better obtained from images that are classified as relevant, but that were close to be classified as irrelevant. These images are ranked based on their *optimum-path costs* in the forest with respect to positive and negative prototypes. This strategy differs from the original definition [13], which uses relevant and irrelevant images. Our strategy reduces the number of false positives, which tends to be significantly higher than the number of false negatives, improving effectiveness. It also considerably reduces the number of images to be ranked, improving efficiency.

A drawback in the planned paradigm is that the user does not know in advance how many iterations would be necessary. However, this could be learned for a given application. Besides, it is also not clear in the greedy paradigm that the system will be able to learn faster than the number of iterations specified in a given planned method. Actually, we are presenting an example where the planned paradigm outperforms in effectiveness the greedy paradigm for a same pattern classification model (OPF). In both paradigms, the minimum number of response images per iteration may also change for distinct queries and users.

Most schemes based on relevance feedback use the greedy paradigm. Fig. 2 presents three examples of simple greedy techniques [14,15]. In Fig. 2a, the positive examples (relevant images) from a first iteration are used to move the next query point to their geometric center in the feature space. This idea stemmed from Rocchio's formula [16] used in document retrieval systems and it has been successfully exploited in CBIR systems [17,18,1,19]. Two other methods use the relevant images as next query points and, depending on the distance to this multi-point query set, different iso-surfaces are formed in the feature space (Figs. 2b and c).

The planned method proposed by Tong and Chang [13] outperforms simple greedy techniques. Some studies [20–22] have reported improvements in Tong and Chang's approach, but it is still the best option to serve as baseline. Hoi et al. [23,24] have also observed a problem with small training sets in Tong and Chang's method [13] and have proposed the use of labeled and unlabeled images in the training set to improve performance. The idea seems interesting for further investigation and can be easily incorporated in our approaches. However, this was not necessary in the present study.

This paper is organized as follows. Section 2 reviews the OPF model and presents the active learning algorithms using the OPF-based greedy and planned paradigms. The experiments and results using five heterogeneous image collections are described in Section 3. Section 4 states the conclusions and discusses our future work.

2. Active learning using optimum-path forest classification

Let $\mathcal{Z}$ be an image database, such that each image $t \in \mathcal{Z}$ is represented by a feature vector $\vec{v}(t)$, computed by a feature extraction function v. The similarity between images $s,t \in \mathcal{Z}$ is measured by a distance function $d(s,t)$. A pair (v,d) is called a *descriptor*. In the case of multiple descriptors encoding shape, color and texture properties, it is possible to combine their distance functions into a composite distance function, as proposed in [7]. Therefore, with no loss of generality, we will describe and evaluate the methods using a single descriptor (v,d).

The problem of interest in CBIR consists of returning a list $\mathcal{X}$ with the N most relevant images in $\mathcal{Z}$ according to the user's opinion with respect to a given query image q. The simplest approach is to return the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q,t)$. However, due to the limitation of (v,d) in representing the user's expectation, this approach very often presents a *semantic gap*, such that the list $\mathcal{X}$ contains relevant and irrelevant images according to the user's opinion. Active learning approaches have been proposed to circumvent the semantic gap problem, by taking the user's feedback about the relevance of the returned images during a few iterations. The user indicates which images are relevant (irrelevant) in $\mathcal{X}$, forming a *labeled training set* $\mathcal{T}$ which gains new elements at each iteration of relevance feedback by $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$.

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

In the methods described here, set $\mathcal{T}$ is used to design an optimum-path forest classifier (OPF) [11]. This process consists of first estimating representative samples (*prototypes*) in $\mathcal{T}$ for each class of images, forming the sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of relevant and irrelevant prototypes, respectively. The training process considers a complete graph, whose nodes are all elements in $\mathcal{T}$ and arcs (s,t) are weighted by $d(s,t)$. Every path in the graph has a cost and minimum-cost paths are computed from $\mathcal{S}_R \cup \mathcal{S}_I$ to each node $t \in \mathcal{T}$, such that the classifier is an optimum-path forest rooted in $\mathcal{S}_R \cup \mathcal{S}_I$. In this forest, the nodes $t \in \mathcal{T} \setminus \mathcal{S}_R \cup \mathcal{S}_I$ are conquered and labeled (in the same class as relevant/irrelevant) by the prototype in $\mathcal{S}_R \cup \mathcal{S}_I$ which offers the optimum path with terminus t. Afterward, this classifier evaluates the images in $\mathcal{Z} \setminus \mathcal{T}$, by computing the cost of the optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$ to each node $t \in \mathcal{Z} \setminus \mathcal{T}$ in an incremental way, and inserts t in a reduced set $\mathcal{Y} \subset \mathcal{Z} \setminus \mathcal{T}$ of *relevant candidates* whenever the optimum path is rooted in $\mathcal{S}_R$.

In the greedy paradigm, the system returns a new list $\mathcal{X} \subset \mathcal{Y}$ with the N closest images in the increasing order of a normalized mean distance $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ between t and the two sets of prototypes (Eq. (3)). We have observed in [10] that the reduced set $\mathcal{Y}$ considerably improves the quality of the subset $\mathcal{X}$ in number of relevant images. The loss of relevant images in $\mathcal{Z} \setminus \mathcal{Y}$ is despicable (e.g., it is on average less than 1.5%). Again, the user may indicate relevant and irrelevant images in $\mathcal{X}$ and a new training set is created by $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ to redesign an improved classifier. This process can be repeated until the user is satisfied.

The basic difference with respect to the planned paradigm is an internal loop of I iterations, which speeds up learning by returning in $\mathcal{X}$, at each iteration, the N *most informative* images of $\mathcal{Y}$. These images are presented in the increasing order of the absolute optimum-path cost difference with respect to $\mathcal{S}_R$ and $\mathcal{S}_I$ (Eq (6)). After the I-th iteration, the images are finally presented in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$.

Although the number of required iterations seems to be higher in the planned approach, this is not necessarily true, as we will see in Section 3. The algorithms involved in each part of these methods are presented in more details next. Section 2.1 describes the OPF training and classification processes. The OPF-based active learning processes using the greedy and planned paradigms are presented in Sections 2.2 and 2.3, respectively.

2.1. Training and classification by optimum-path forest

Given a complete graph, whose nodes are all images in $\mathcal{T}$, a path π_t in the graph with terminus t is a sequence $\langle t_1,t_2,\ldots,t_n=t\rangle$ of distinct nodes. The *strength of connectedness* between the origin $R(\pi_t)=t_1$ and the terminus $t_n=t$ is inversely proportional to maximum arc weight $d(t_i,t_{i+1})$ (i.e., weakest link) along it, as defined by

$$c(\pi_t) = \max_{i=1,2,\ldots,n-1}\{d(t_i,t_{i+1})\}. \quad (1)$$

A path π_t is optimum when $c(\pi_t) \le c(\tau_t)$ for any other path τ_t with the same terminus t. For every node $t \in \mathcal{T}$, we wish to compute a minimum-cost path with terminus t and origin $R(t) \in \mathcal{S}_R \cup \mathcal{S}_I$. The idea is to obtain an optimum partition of $\mathcal{T}$ such that the relevant and irrelevant prototypes in $\mathcal{S}_R$ and $\mathcal{S}_I$ will compete with each other and every training node $t \in \mathcal{T}$ will be assigned to the class of its most strongly connected prototype. This partition is computed as an optimum-path forest P—an acyclic graph which stores all optimum paths in a predecessor map P, i.e., a function which assigns to every node $t \in \mathcal{T} \setminus \mathcal{S}_R \cup \mathcal{S}_I$ its predecessor node $P(t)$ in the optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$, or a mark $P(t)=nil$ when $t \in \mathcal{S}_R \cup \mathcal{S}_I$.

Algorithm 1 presents the computation of an optimum-path forest P for path-cost function c [25,11] (Eq. (1)) constrained to start in $\mathcal{S}_R \cup \mathcal{S}_I$. It outputs for each node $t \in \mathcal{T}$, the minimum cost $C(t)$ of the optimum path with terminus t, its root node $R(t) \in \mathcal{S}_R \cup \mathcal{S}_I$, and the predecessor $P(t)$ in that optimum path. A list $\mathcal{T}'$ of the training nodes in the increasing order of cost $C(t)$ is also output. It is used to speedup classification.

Given that the optimum paths for function c tend to choose the closest nodes to form each link along the paths, the sets of prototypes $\mathcal{S}_R$ and $\mathcal{S}_I$ must be chosen from the closest samples between the relevant and irrelevant classes. These prototypes tend to avoid misclassification by an optimum path coming from the opposite class. Let $\lambda(t)$ be the class of image $t \in \mathcal{T}$, which may be relevant or irrelevant. The prototypes are obtained by computing a *minimum spanning tree* (MST) in the complete graph with nodes in $\mathcal{T}$ [26]. For each arc (s,t) in the MST, if $\lambda(s) \neq \lambda(t)$ then s and t are marked as prototypes. If $\lambda(s)$ is relevant and $\lambda(t)$ is irrelevant, then s is inserted in $\mathcal{S}_R$ and t is inserted in $\mathcal{S}_I$. Similarly, it is done in the other way around.

The algorithm follows the dynamic programming scheme [25]. Lines 1–7 initialize the cost, predecessor, and root maps, forcing the optimum paths to start in $\mathcal{S}_R \cup \mathcal{S}_I$, and insert the roots in Q. The main loop computes an optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$ to every node $s \in \mathcal{T}$ in a non-decreasing order of cost (Lines 8–20). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q, preserving its order in $\mathcal{T}'$ (Line 9). The rest of the lines evaluate if the path that reaches an adjacent node $t \neq s$ through s is cheaper than the current path with terminus t and update Q, $C(t)$, $R(t)$ and $P(t)$ accordingly.

Algorithm 1. OPF Algorithm.

Input: Training set $\mathcal{T}$, the sets of prototypes $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$, and a descriptor (v,d).
Output: Optimum-path forest P (predecessor map), path-cost map C, root map R, and the ordered list $\mathcal{T}'$ of training nodes.
Auxiliary: Priority queue Q and cost variable cst.

```
1  for each s ∈ T \ S_R ∪ S_I do
2    Set C(s) ← +∞
3  end
4  for each s ∈ S_R ∪ S_I do
5    Set C(s) ← 0, P(s) ← nil,
6    R(s) ← s, and insert s into Q.
7  end
8  while Q is not empty do
9    Remove from Q a node s with minimum C(s) and insert s
     in T'.
10   for each t ∈ T such that C(t) > C(s) do
11     Compute cst ← max{C(s),d(s,t)}
12     if cst < C(t) then
13       if C(t) ≠ +∞ then
14         remove t from Q.
15       end
16       P(t) ← s, R(t) ← R(s) and C(t) ← cst.
17       Insert t in Q,
18     end
19   end
20 end
```

In the classification phase, for every sample $t \in \mathcal{Z} \setminus \mathcal{T}$, the optimum path with terminus t can be easily identified by finding which training node $s^* \in \mathcal{T}$ provides the minimum value in

$$C(t) = \min_{\forall s \in \mathcal{T}}\{\max\{C(s),d(s,t)\}\}. \quad (2)$$

Node s^* is the predecessor $P(t)$ in the optimum path with terminus t and the image t is classified as $\lambda(R(s^*))$.

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

The role of $\mathcal{T}'$ in the above algorithm is to speed up the evaluation of Eq. (2) which can halt when $\max\{C(s),d(s,t)\} < C(p)$, for a node p whose position in $\mathcal{T}'$ succeeds the position of s [27].

2.2. Greedy learning using OPF

Algorithm 2 presents our greedy active learning approach using OPF (GOPF) [10]. It returns a list $\mathcal{R} \subset \mathcal{Z}$ of images in the decreasing order of relevance. In Line 1, this list and the training set $\mathcal{T}$ are initialized to empty sets. For a given query image q and image database $\mathcal{Z}$, the algorithm first returns a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q,t)$ (Line 2). The learning process is executed in the main loop (Lines 3–9). In Line 4, the user marks the relevant images in $\mathcal{X}$, and the remaining images are considered as irrelevant. This essentially creates a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ where each image $t \in \mathcal{T}$ has a relevance label $\lambda(t)$. The relevant images of $\mathcal{X}$ are inserted in the output list $\mathcal{R}$ (Line 5). The OPF training is computed in Line 6 (Algorithm 1). It results sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of relevant and irrelevant prototypes; the OPF attributes, predecessor $P(t)$, cost $C(t)$, and root $R(t)$ for each $t \in \mathcal{T}$; and set $\mathcal{T}'$ of training images in the increasing order of optimum cost $C(t)$ for faster classification. In Line 7, OPF is used to classify images in $\mathcal{Z} \backslash \mathcal{T}$ using Eq. (2), forming a set $\mathcal{Y}$ with the images classified as relevant. A new list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ is created in the increasing order of a *normalized mean distance* $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ between t and the two sets of prototypes:

$$\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I) = \frac{\overline{d}(t,\mathcal{S}_R)}{\overline{d}(t,\mathcal{S}_R)+\overline{d}(t,\mathcal{S}_I)}, \tag{3}$$

$$\overline{d}(t,\mathcal{S}_R) = \frac{1}{|\mathcal{S}_R|} \sum_{\forall s \in \mathcal{S}_R} d(s,t), \tag{4}$$

$$\overline{d}(t,\mathcal{S}_I) = \frac{1}{|\mathcal{S}_I|} \sum_{\forall s \in \mathcal{S}_I} d(s,t). \tag{5}$$

The main loop repeats until the user is satisfied and the retrieved images in $\mathcal{R}$ are all relevant images together with the last set $\mathcal{X}$ in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$.

Algorithm 2. GOPF Algorithm.

Input: A query image q, a descriptor (v,d), the number N of returned images per iteration, and an image database $\mathcal{Z}$.
Output: A list $\mathcal{R} \subset \mathcal{Z}$ of retrieved images in the decreasing order of relevance.
Auxiliary: Set $\mathcal{T}$ of training images, maps $(R,P,C,\mathcal{T}')$ of the OPF classifier, sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of prototypes, set $\mathcal{Y} \subset \mathcal{Z} \backslash \mathcal{T}$ of images classified as relevant, and ordered list $\mathcal{X} \subset \mathcal{Y}$ of N returned images per iteration.

1 Set $\mathcal{R} \leftarrow \emptyset$ and $\mathcal{T} \leftarrow \emptyset$.
2 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q,t)$.
3 **while** *the user is not satisfied* **do**
4 The user marks relevant (irrelevant) images, forming a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ where each image has a relevance label $\lambda(t)$.
5 Insert in $\mathcal{R}$ the relevant images of $\mathcal{X}$.
6 Compute sets $\mathcal{S}_R$ and $\mathcal{S}_I$ of prototypes in $\mathcal{T}$ (Section 2.1) and execute $(R,P,C,\mathcal{T}') \leftarrow \text{OPF}(\mathcal{T},\mathcal{S}_R,\mathcal{S}_I,v,d)$ (Algorithm 1).
7 Classify images in $\mathcal{Z} \backslash \mathcal{T}$ using $(R,P,C,\mathcal{T}')$ and $\lambda(t)$ for $t \in \mathcal{S}_R \cup \mathcal{S}_I$ (Eq. (2)), and create set $\mathcal{Y}$ with the relevant candidates.
8 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$.
9 **end**
10 Return $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

2.3. Planned learning using OPF

Algorithm 3 presents our planned learning approach using OPF (POPF). It also returns a list $\mathcal{R} \subset \mathcal{Z}$ of images in the decreasing order of relevance. Lines 1 and 2 are the same as in Algorithm 2 and the learning process is also executed in the main loop (Lines 3–13). The basic difference between them is the internal loop from Lines 5 to 11, which forces I iterations (Line 4) of learning. Lines 6–9 do essentially the same done by Algorithm 2 in Lines 4–7, which results in a list $\mathcal{Y} \subset \mathcal{Z} \backslash \mathcal{T}$ of images classified as relevant. However, Line 10 creates a list $\mathcal{X}$ with the N *most informative* images among the relevant candidates in $\mathcal{Y}$. The most informative images are the most likely to become prototypes in $\mathcal{S}_R \cup \mathcal{S}_I$, so they speed up the learning process. They are also the most difficult images to be assigned to any class, because they are in the boundary between classes. Therefore, the most informative images in $\mathcal{X}$ are presented in the increasing order of the absolute cost difference $d_c(t,\mathcal{S}_R,\mathcal{S}_I)$, considering all $t \in \mathcal{Y}$:

$$d_c(t,\mathcal{S}_R,\mathcal{S}_I) = |C_R(t) - C_I(t)|, \tag{6}$$

where $C_R(t)$ is the cost of the best path with root in $\mathcal{S}_R$ and $C_I(t)$ is the cost of the best path with root in $\mathcal{S}_I$. At least one of them is the optimum path with terminus t. The ordered list $\mathcal{T}'$ is used to obtain $C_R(t)$ and $C_I(t)$ by constraining the search for predecessor nodes $s \in \mathcal{T}'$ whose root $R(s)$ is either in $\mathcal{S}_R$ or in $\mathcal{S}_I$, respectively:

$$C_R(t) = \min_{\forall s \in \mathcal{T} | R(s) \in \mathcal{S}_R} \{\max\{C(s),d(s,t)\}\}, \tag{7}$$

$$C_I(t) = \min_{\forall s \in \mathcal{T} | R(s) \in \mathcal{S}_I} \{\max\{C(s),d(s,t)\}\}. \tag{8}$$

After the I-th iteration, a list $\mathcal{X}$ is finally presented with the N closest images $t \in \mathcal{Y}$ in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ (Line 12). The whole process may be repeated until the user is satisfied and the retrieved images in $\mathcal{R}$ are all relevant images together with the last set $\mathcal{X}$ in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ (Line 14).

Algorithm 3. POPF Algorithm.

Input: A query image q, a descriptor (v,d), the number N of returned images per iteration, and an image database $\mathcal{Z}$.
Output: A list $\mathcal{R} \subset \mathcal{Z}$ of retrieved images in the decreasing order of relevance.
Auxiliary: Set $\mathcal{T}$ of training images, maps $(R,P,C,\mathcal{T}')$ of the OPF classifier, sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of prototypes, set $\mathcal{Y} \subset \mathcal{Z} \backslash \mathcal{T}$ of images classified as relevant, ordered list $\mathcal{X} \subset \mathcal{Y}$ of N returned images per iteration, and variables i and I for the planned iterations.

1 Set $\mathcal{R} \leftarrow \emptyset$ and $\mathcal{T} \leftarrow \emptyset$.
2 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q,t)$.
3 **while** *the user is not satisfied* **do**
4 Ask the user for the number I of planned iterations.
5 **for** $i=1$ *to* I **do**
6 The user marks relevant (irrelevant) images, forming a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ where each image has a relevance label $\lambda(t)$.
7 Insert in $\mathcal{R}$ the relevant images of $\mathcal{X}$.
8 Compute sets $\mathcal{S}_R$ and $\mathcal{S}_I$ of prototypes in $\mathcal{T}$ (Section 2.1) and execute training by $(R,P,C,\mathcal{T}') \leftarrow \text{OPF}(\mathcal{T},\mathcal{S}_R,\mathcal{S}_I,v,d)$ (Algorithm 1).
9 Classify images in $\mathcal{Z} \backslash \mathcal{T}$ using $(R,P,C,\mathcal{T}')$ and $\lambda(t)$ for $t \in \mathcal{S}_R \cup \mathcal{S}_I$ (Eq. (2)), and create set $\mathcal{Y}$ with the relevant candidates.
10 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ in the increasing order of $d_c(t,\mathcal{S}_R,\mathcal{S}_I)$ (Eq. (6)).
11 **end**
12 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ in the increasing order of $\overline{d}(t,\mathcal{S}_R,\mathcal{S}_I)$.

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

13 **end**
14 Return $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

3. Experiments and results

In any real world application of CBIR, it is very important for a method to minimize the average response time per query and the average number of marked images per query, while it maximizes the average degree of user satisfaction. These challenges will also include the choice of suitable and effective image descriptors for such an application. Given that, we are not addressing these issues here, the experiments in this section aim to give us at least a strong indication of which among the methods should be our first choice for further investigation in the context of a given CBIR application. For a same image descriptor and distinct data sets, we are evaluating the flexibility of the methods to deal with different applications. The efficiency issues are related to the average computational time per query, which we are measuring for each method, and the choice of small numbers N (number of response images per relevance feedback iteration) and I (number of iterations). The average number of marked images per query is related to the product $N \times I$. We have then chosen $N=30$, also because the 30 images fit on the computer screen and this would facilitate their visual verification in practical applications. The methods have been evaluated with I equal to 3, 5, and 8 iterations for Corel database and I equal to 3 and 8 for the other databases.

It should be clear that both approaches, GOPF and POPF, can be used with indexing structures and techniques that combine multiple descriptors. Therefore, we evaluate the methods in this section by using a same color descriptor and a set of five heterogeneous image collections. We use the BIC descriptor [4] and the following image collections, which represent distinct challenges for CBIR systems:

- Caltech 101 [28]: Images of 9144 objects belonging to 101 categories—about 40–800 per category. Most categories have about 50 images.
- Coil-100 [29]: Images of 100 objects. Pictures of each object were taken in 72 different poses, totalizing 7200 images.
- Corel [30]: A subset of 3906 images from the Corel GALLERY Magic–Stock Photo Library 2. The images were pre-classified into 85 classes. Each class has a different number of images varying from 7 to 98.
- MSRCORID [31]: A database with 4320 images grouped into 20 classes—from 36 to 652 images per class. Most of the classes has about 200 images.
- PASCAL [32]: This database consists of images from Flickr.[1] We use a subset with 3448 images grouped into 23 classes. Each class has a different number of images, varying from 72 to 446 sub-images (regions of interest which may come from a same image).

Besides GOPF and POPF, we have chosen two baseline approaches for the experiments: the query expansion method (QEX) [12] and the SVM-based method by Tong and Chang (SAL) [13]. They all use the same initial queries q, number N of returned images per iteration, and, for each initial q, they start from a same list $\mathcal{X} \subset \mathcal{Z}$ with N images $t \in \mathcal{Z}$ in the increasing order of distance $d(q,t)$—in the BIC descriptor, $d(q,t)$ is the logarithm of the L1 distance (dLog). Their main loop stops when the user is satisfied, but we will fix the number I of iterations for the purpose of comparison. Thus, greedy methods stop after I iterations of the main loop and planned methods perform a single iteration of the main loop with I iterations of the internal loop.

QEX is a greedy approach selected to illustrate the importance of using irrelevant images in multi-point query. QEX does not use irrelevant points, and both GOPF and POPF are multi-point query methods which rank images by their distances with respect to the sets of relevant and irrelevant prototypes. In QEX, the user marks relevant images in $\mathcal{X}$, forming a set $\mathcal{R} \subset \mathcal{X}$ of relevant images. The images in $\mathcal{R}$ are clustered and the centroids of the clusters are used as the next query points q_i, $i = 1,2,\ldots$. The N closest images $t \in \mathcal{Z} \backslash \mathcal{R}$ are ranked in the increasing order of distance $d(t,q_i)$ to their closest query point q_i in order to form a new set $\mathcal{X}$, as illustrated in Fig. 2c. The result after I iterations is presented in $\mathcal{R} \cup \mathcal{X}$, being $\mathcal{X}$ the last set of N returned images.

SAL is a planned approach which uses SVM classifiers rather than OPF. It is also known as SVM_{ACTIVE} or SVM_{AL} in the literature. It was chosen because it can be considered the state-of-the-art in planned learning. As in POPF, the user marks relevant and irrelevant images in $\mathcal{X}$, forming a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$. Training consists of finding support vectors in $\mathcal{T}$, applying a Gaussian kernel to obtain a new feature space, and computing the optimum hyperplane that separates relevant and irrelevant images of $\mathcal{T}$ in that feature space. During the I iterations of the internal loop, SAL returns in $\mathcal{X}$ the N closest images $t \in \mathcal{Z} \backslash \mathcal{T}$ to the optimum hyperplane (i.e., images from both sides), which represent the most informative images to improve training in a next iteration. After the I-th iteration, SAL returns in $\mathcal{X}$ the N farthest images $t \in \mathcal{Z} \backslash \mathcal{T}$ to the optimum hyperplane on its relevant side.

Precision–recall curves are used to measure the effectiveness of all methods in the returned set $\mathcal{R}$ after the I-th iteration. For each image database, we simulate the user behavior by using each image $q \in \mathcal{Z}$ as initial query point and marking the relevant points (images from the same class of the query) from $N=30$ returned images in the list $\mathcal{X}$ of each iteration.

Figs. 3–5 present the mean precision–recall curves of each method (POPF, GOPF, SAL, and QEX) for $I=3$, 5, 8 iterations of active learning using the Corel database. It is possible to observe that the new approach, POPF, outperformed all methods for any number I of iterations. Both OPF-based approaches, GOPF and POPF, started with good performance, but POPF evolves faster (learning in less number of iterations) than GOPF. This result is similar in Figs. 6–13, which present the mean precision–recall curves of each method for $I=3$, 8 iterations. This demonstrates that the planned paradigm can outperform the greedy paradigm in effectiveness. It is very likely that the user would be satisfied at this point in a practical situation. The other methods presented at least 10% less precision on average at 30% of recall for three iterations.

The gain in effectiveness of POPF over GOPF, and of both OPF-based approaches over the other methods is also clear in the remaining databases. Figs. 6–13 show the mean precision–recall curves of each method in the image databases Caltech, Coil-100, MSRCORID, and Pascal, respectively, for three and eight iterations of active learning. We consider three iterations a reasonable number for practical situations and the OPF-based approaches present considerable advantages in this case. It is possible to observe that Caltech database is the most difficult one for the BIC descriptor and Coil-100 database is the easiest one. However, QEX outperformed SAL in Coil-100 database and the same happened for three iterations in Pascal database. Indeed, SAL usually requires more iterations to improve the performance and this might be the reason that its recent advances include unlabeled images in the training set [24].

We have also evaluated the efficiency of all methods QEX, SAL, GOPF and POPF. Table 1 shows their total execution times for eight iterations of active learning in each database. Their average

[1] www.flickr.com

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

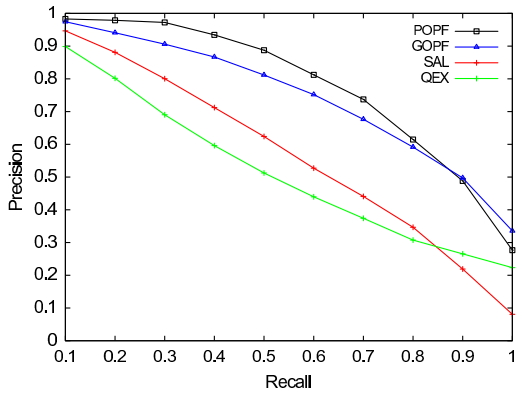


Fig. 3. Mean precision–recall curves in *Corel* database, third iteration.

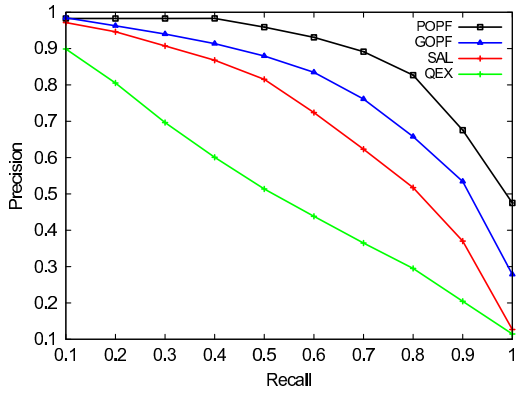


Fig. 4. Mean precision–recall curves in *Corel* database, fifth iteration.

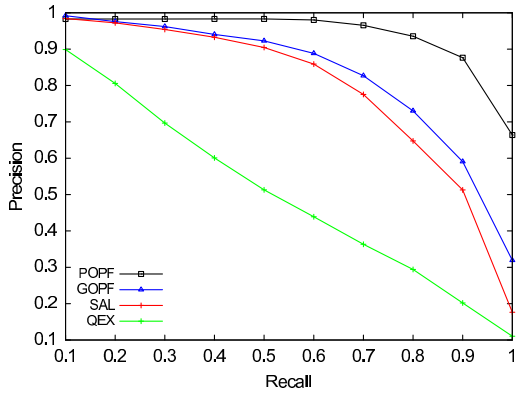


Fig. 5. Mean precision–recall curves in *Corel* database, eighth iteration.

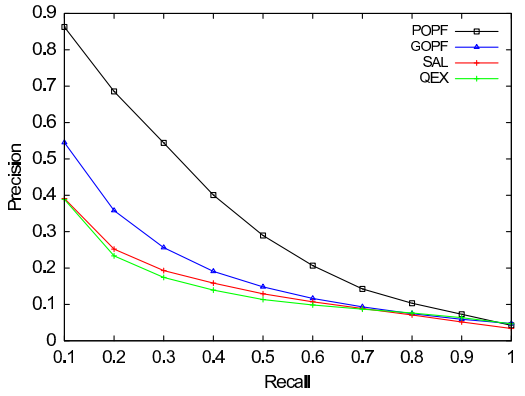


Fig. 6. Mean precision–recall curves in *Caltech* database for three iterations.

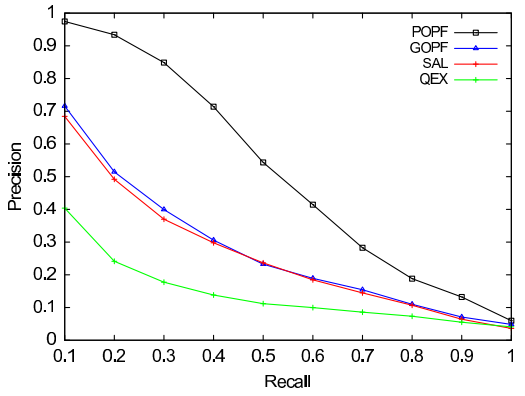


Fig. 7. Mean precision–recall curves in *Caltech* database for eight iterations.

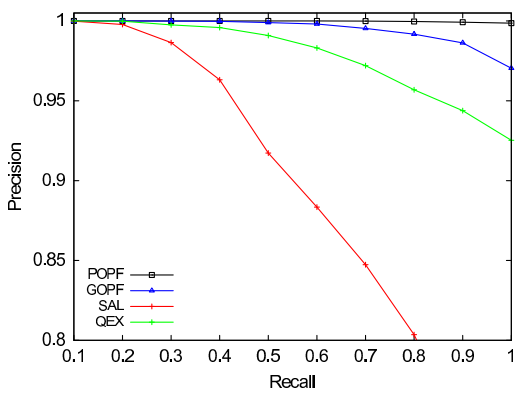


Fig. 8. Mean precision–recall curves in *Coil-100* database for three iterations.

time per query is shown in Table 2. It should be clear that SAL is the most expensive approach, due to the SVM training, and that QEX and the OPF-based methods provide interactive response times. Note that we are not using indexing structures to retrieve images from $\mathcal{Z}$. We will certainly need them for huge databases, but even considering the used database sizes, the results were very satisfactory. In Corel database, for instance, GOPF and POPF take about 0.1 s to present images at the eighth iteration while SAL takes 8.9 s in average. This confirms the efficiency gains of OPF over SVM, as reported before [11]. The tests were performed in a machine with Intel Core i7 processor at 2.8 GHz and 8 GB RAM, running the Linux operational system.

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

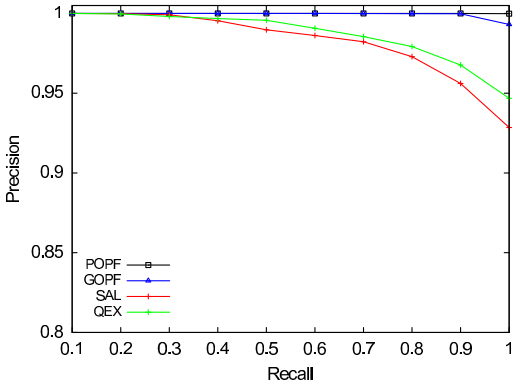


Fig. 9. Mean precision–recall curves in *Coil-100* database for eight iterations.

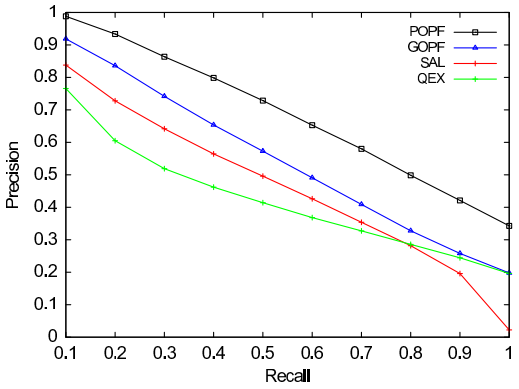


Fig. 10. Mean precision–recall curves in *MSRCORID* database for three iterations.

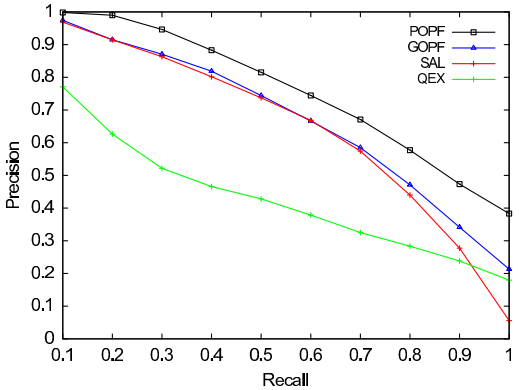


Fig. 11. Mean precision–recall curves in *MSRCORID* database for eight iterations.

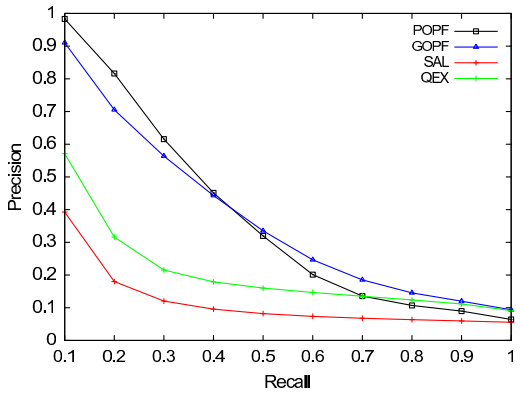


Fig. 12. Mean precision–recall curves in *Pascal* database for three iterations.

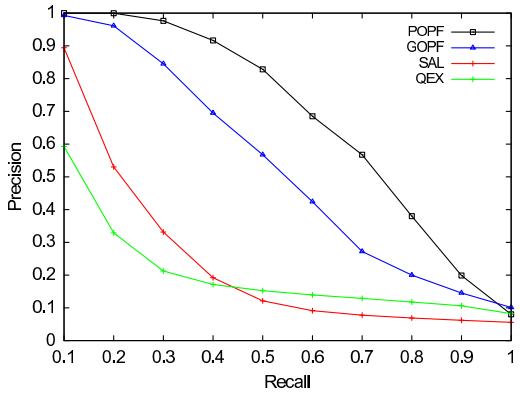


Fig. 13. Mean precision–recall curves in *Pascal* database for eight iterations.

Table 1
Total execution time for eight iterations (min).

Database	Caltech	Coil-100	Corel	MSRCORID	PASCAL
QEX	169.5	184.3	54.7	60.5	25.7
SAL	7229.8	4608.0	3312.3	3749.8	2744.6
GOPF	178.0	101.7	37.4	58.1	35.8
POPF	178.0	101.8	37.5	58.2	35.9

Table 2
Average execution time per query (s).

Database	Caltech	Coil-100	Corel	MSRCORID	PASCAL
QEX	0.14	0.20	0.11	0.06	0.05
SAL	5.93	4.80	6.36	6.51	5.97
GOPF	0.15	0.11	0.07	0.10	0.08
POPF	0.15	0.11	0.07	0.10	0.08

4. Conclusion and future work

We have discussed greedy and planned active learning paradigms for CBIR systems, being both of them based on the optimum-path forest classifier (GOPF and POPF for greedy and planned methods, respectively). GOPF was proposed previously [10], but it was evaluated in this work with more databases. POPF is a new approach never presented before.

Our experiments involved a reasonable amount of user interaction from the practical point of view, by setting low numbers of response images and iterations; five databases, representing distinct levels of challenges for CBIR systems; and four methods,

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

POPF, GOPF [10], a simple query expansion (QEX) [12], and the SVM-based method, as proposed by Tong and Chang (SAL) [13].

GOPF outperformed SAL, demonstrating that a greedy approach can be better than a planned method. However, POPF outperformed all methods, indicating that the planned approach is likely better than the greedy method, when the classification model is the same. The gains of POPF and GOPF in effectiveness and efficiency over SAL can be explained by the use of only relevant candidates for ranking, which has eliminated most false positives, and the computational efficiency of the OPF classifier, which is significantly higher than the efficiency of the SVM classifier. GOPF and POPF also obtained interactive response times. Considering these results under a reasonable amount of user involvement, we may conclude that POPF should be the first choice to investigate real world applications of CBIR.

Our future work involves the use of multiple descriptors and techniques to combine them. We intend to use other descriptors based on shape, texture and color and combine them by using techniques such as Bayesian framework, genetic programming [7] and other similar approaches. We also intend to investigate the use of unlabeled images in the training sets and to evaluate the OPF-based methods for multiple users.

Acknowledgments

The first author thanks CNPq for financial support (140968/2007-5). The second author thanks CNPq (481556/2009-5, 302617/2007-8) and FAPESP (2007/52015-0, 2008/57428-4).

References

[1] R.S. Torres, A.X. Falcão, Content-based image retrieval: theory and applications, Revista de Informática Teórica e Aplicada 13 (2) (2006) 161–185.
[2] P. Ciaccia, M. Patella, P. Zezula, M-tree: an efficient access method for similarity search in metric spaces, in: Proceedings of the 23rd VLDB International Conference1997, pp. 426–435.
[3] M.R. Vieira, C. Traina Jr., F.J.T. Chino, A.J.M. Traina, Dbm-tree: a dynamic metric access method sensitive to local density data, Journal of Information and Data Management 1 (1) (2010) 111–128.
[4] R.O. Stehling, M.A. Nascimento, A.X. Falcão, A compact and efficient image retrieval approach based on border/interior pixel classification, in: CIKM '02: Proceedings of the Eleventh International Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2002, pp. 102–109 〈http://doi.acm.org/10.1145/584792.584812〉.
[5] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Found. Trends. Comput. Graph. Vis. 3 (3) (2008) 177–280. 〈http://dx.doi.org/10.1561/0600000017〉.
[6] R. Ohbuchi, Y. Hata, Combining multiresolution shape descriptors for 3d model retrieval, in: Proc. WSCG 2006, Plzen, Czech Republic, 2006.
[7] R. Torres, A. Falcão, M. Gonçalves, J. Papa, B. Zhang, W. Fan, E. Fox, A genetic programming framework for content-based image retrieval, Pattern Recognition 42 (2) (2009) 217–312.
[8] H. Lejsek, F. Ásmundsson, B. Jónsson, L. Amsaleg, An efficient disk-based index for approximative search in very large high-dimensional collections, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (5) (2008) 869–883.
[9] E. Valle, M. Cord, S. Philipp-Foliguet, High-dimensional descriptor indexing for large multimedia databases, in: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2008, pp. 739–748. 〈http://doi.acm.org/10.1145/1458082.1458181〉.
[10] A.T. Silva, A.X. Falcão, L.P. Magalhães, A new CBIR approach based on relevance feedback and optimum path forest classification, Journal of WSCG 18 (1–3) (2010) 73–80.
[11] J.P. Papa, A.X. Falcão, C.T.N. Suzuki, Supervised pattern classification based on optimum-path forest, International Journal of Imaging Systems and Technology 19 (2) (2009) 120–131.
[12] K. Porkaew, K. Chakrabarti, S. Mehrotra, Query refinement for multimedia similarity retrieval in mars, in: Proceedings of ACM Multimedia1999, pp. 235–238.
[13] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: MULTIMEDIA '01: Proceedings of the Ninth ACM International Conference on Multimedia, ACM, New York, NY, USA, 2001, pp. 107–118 〈http://doi.acm.org/10.1145/500141.500159〉.
[14] D.-H. Kim, C.-W. Chung, Qcluster: relevance feedback using adaptive clustering for content-based image retrieval, in: Proc. of the ACM SIGMOD Int. Conf. on Management of Data2003, pp. 599–610.
[15] D. Liu, K.A. Hua, K. Vu, N. Yu, Fast query point movement techniques for large CBIR systems, IEEE Transactions on Knowledge and Data Engineering 21 (5) (2009) 729–743. 〈http://dx.doi.org/10.1109/TKDE.2008.188〉.
[16] J.J. Rocchio, Relevance Feedback in Information Retrieval, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971 (pp. 313–323).
[17] Y. Rui, T.S. Huang, S. Mehrotra, Content-based image retrieval with relevance feedback in mars, in: Proc. IEEE Int. Conf. on Image Proc.1997, pp. 815–818.
[18] X.S. Zhou, T.S. Huang, Relevance feedback in image retrieval: a comprehensive review, Multimedia Systems 8 (6) (2003) 536–544.
[19] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Computing Surveys 40 (2) (2008) 1–60. 〈http://doi.acm.org/10.1145/1348246.1348248〉.
[20] K.-S. Goh, E.Y. Chang, W.-C. Lai, Multimodal concept-dependent active learning for image retrieval, in: MULTIMEDIA '04: Proceedings of the 12th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, 2004, pp. 564–571. 〈http://doi.acm.org/10.1145/1027527.1027664〉.
[21] N. Panda, K.-S. Goh, E.Y. Chang, Active learning in very large databases, Multimedia Tools and Applications 31 (3) (2006) 249–267.
[22] C.K. Dagli, S. Rajaram, T.S. Huang, Leveraging active learning for relevance feedback using an information theoretic diversity measure, in: ACM Conferece on Image and Video Retrieval (CIVR), 2006, pp. 123–132.
[23] S. Hoi, M. Lyu, A semi-supervised active learning framework for image retrieval, vol. 2, 2005, pp. 302–309 〈http://dx.doi.org/10.1109/CVPR.2005.44〉.
[24] S.C.H. Hoi, R. Jin, J. Zhu, M.R. Lyu, Semisupervised SVM batch mode active learning with applications to image retrieval, ACM Transactions on Information Systems 27 (3) (2009) 1–29. 〈http://doi.acm.org/10.1145/1508850.1508854〉.
[25] A. Falcão, J. Stolfi, R. Lotufo, The image foresting transform: theory, algorithms, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 19–29.
[26] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT, 1990.
[27] J.P. Papa, F.A.M. Cappabianco, A.X. Falcão, Optimizing optimum-path forest classification for huge datasets, in: Proceedings of the 20th International Conference on Pattern Recognition2010.
[28] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: IEEE CVPR 2004, Workshop on Generative-Model, Based Vision2004.
[29] S.A. Nene, S.K. Nayar, H. Murase, Columbia university image library (coil-100) 〈http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php〉.
[30] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 947–963.
[31] M.R. Cambridge, Microsoft research cambridge, object recognition image database 1.0 〈http://research.microsoft.com/vision/cambridge/recognition/〉.
[32] M. Everingham, L.V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results 〈http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/index.html〉.

André Tavares da Silva is a PhD student at University of Campinas (Unicamp). He has experience in computer science, with emphasis on Computer Graphics and Vision, acting on the following topics: Geometric Modeling, Visualization and Content-Based Image Retrieval.

Alexandre Xavier Falcão is an associate professor at University of Campinas. He has experience in computer science, with emphasis on Image Processing, acting on the following topics: Image Processing, Visualization and Analysis, Content-Based Image Retrieval, Pattern Recognition and Machine Learning, Digital Video and Biomedical Imaging Applications.

Léo Pini Magalhães is a full professor at University of Campinas. He has experience in computer science, with emphasis on Computer Graphics, mainly in the following areas: Animation, Image Synthesis and Coordination in computer environments.

Please cite this article as: A.T. da Silva, et al., Active learning paradigms for CBIR systems based on optimum-path forest classification, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.04.026

Interactive Classification of Remote Sensing Images by using Optimum-Path Forest and Genetic Programming

Jefersson Alex dos Santos[1], André T. da Silva[2], Ricardo da S. Torres[1], Alexandre X. Falcão[1], Léo P. Magalhães[2], and Rubens A. C. Lamparelli[3]

[1] Institute of Computing
[2] School of Electrical and Computer Engineering
[3] Center for Research in Agriculture
University of Campinas, Campinas, SP, Brazil
jsantos@ic.unicamp.br,atavares@dca.fee.unicamp.br,rtorres@ic.unicamp.br, afalcao@ic.unicamp.br,leopini@fee.unicamp.br,rubens@cpa.unicamp.br,

Abstract. The use of remote sensing images as a source of information in agribusiness applications is very common. In those applications, it is fundamental to know how the space occupation is. However, identification and recognition of crop regions in remote sensing images are not trivial tasks yet. Although there are automatic methods proposed to that, users sometimes prefer to identify regions manually. That happens because these methods are usually developed to solve specific problems, or, when they are of general purpose, they do not yield satisfying results. This work presents a new interactive approach based on relevance feedback to recognize regions of remote sensing. Relevance feedback is a technique used in content-based image retrieval (CBIR) tasks. Its objective is to aggregate user preferences to the search process. The proposed solution combines the Optimum-Path Forest (OPF) classifier with composite descriptors obtained by a Genetic Programming (GP) framework. The new approach has presented good results with respect to the identification of pasture and coffee crops, overcoming the results obtained by a recently proposed method and the traditional Maximimun Likelihood algorithm.

Keywords: Remote Sensing Image Classification; Genetic Programming; Optimum-Path Forest; Relevance Feedback

1 Introduction

Agriculture productivity is strongly dependent on monitoring and planning activities. Production estimation and land use are the basis for government policies to finance agricultural activities. Thus, there is a huge demand for information systems that allow to store, analyze, and handle geographic data. This is the purpose of Geographic Information Systems (GISs). Most of existing GIS-based applications rely on the use of Remote Sensing Images (RSIs) to crop monitoring.

Because RSIs are raster, to obtain vectorial information is necessary to extract regions of interest. In addition to the typical problems in pattern recognition research, identifying crop areas in RSIs faces hard problems associated, for instance, with terrain distortions. What is more, RSIs, contrary to common images, do not encode just human visible information, but also other spectral bands (for example, infrared). For this reason, the recognition task normally needs classification strategies which exploit RSI properties related to both spectral and texture patterns.

The process of recognizing regions is called classification and can be done both automatically or manually. Sometimes users prefer to identify regions manually because the results of automatic approaches are unsatisfactory. The most successful RSI classification methods are normally created to a specific target or data [14]. General purpose methods, however, are very sensitive to noise. Furthermore, the spectral response and the texture patterns observed for a given crop can be different. A crop can be planted in different ways and this factor, allied to the different phases of plants, tends to create distinction between regions where the same culture is being cultivated. Therefore, in practical situations, the results of automatic methods need to be revised.

This work aims to present a new interactive approach for classifying regions in RSIs. The proposed solution relies on the use of an interactive strategy, called *relevance feedback* (RF), based on which the classification system can learn what regions are of interest, given what is indicated by users. The proposal is a new hybrid method, named *GOPF*, which uses a GP framework to create composite image descriptors [5] and the optimum-path forest (OPF) classifier [11] to determine regions of interest. OPF is a classification method which represents each class of objects by one or more optimum-path trees rooted at given samples, called prototypes.

2 Related Work

In [9], Lu & Weng present an overview of the problem of classification of remote sensing images including the steps which comprise the process (extraction of features, segmentation, classification and accuracy assessment) and the research challenges faced. For each step, most of the existing techniques until 2005 are presented, grouped by the approach adopted (such as techniques that exploits classification by pixels or regions).

The classification algorithm based on pixels, MaxVer (*Maximum Likelihood Classification*) [12], is still one of the most popular. On the other hand, the growth of classification approaches based on regions, for instance, is analyzed in [3]. The article proves that the growth in the number of new approaches published accompanies the increase of the accessibility to high-resolution images and, hence, the development of alternative techniques to the classification based on pixels.

Apart from the classification methods presented in [14, 12, 3], several others have been proposed recently [8, 13, 1, 10, 2]. The novelty of these approaches re-

lies on: resolution and number of bands of ISRs; type of extracted features; used learning technique, and level of discrimination between the classes of the image (some studies include all the vegetation types in the same class, for example). Li et. al [8] proposed a method based on a regions adjacency graph for segmentation of images with high resolution. The regions are segmented according to the combination of shape, color and texture features. The RSIs classification methods proposed by Munoz-Mari et al. [10] and Basi & Melgani [2] are based on Support vector machines (SVMs). The first [10] proposes a supervised classification method called SVDD. The experiments were made by differentiating various classes of vegetation (corn, grass, pasture, trees, etc.). As far as [2] is concerned, they proposed an RSIs classification system based on SVM in which Genetic Algorithms (GA) are used to find the best set of parameters of SVM. The extracted features are based on pixels. Low-resolution aerial images were used. Besides [2], both RSIs classification methods proposed by Tseng et. al [13] and Bandyopadhyay et. al [1] use Genetic algorithms (GA). The former uses GA to find the configuration parameters of a neural network while the latter uses GA for clustering the pixels of the RSIs. Although both use the pixel information, in [13], the vegetation classes are distinguished and images of middle and high resolution are used.

The main advantages of the proposed method against the afore mentioned approaches are the use of a runtime technique for combination of features (genetic programming) and the refinement of the OPF-based classification system by taking into account the user interaction. Moreover, most of the mentioned works do not address the problem of specific crops recognition. They group different classes into larger sets or even a single one.

We have recently proposed an interactive method for classification of remote sensing images based on Genetic Programming, GP_{SR} [6]. That method allows users to interact with the classification system, indicating regions of interest (and those which are not). This feedback information is employed by a genetic programming approach for learning user preferences and combining image region descriptors that encode spectral and texture properties. One remarkable advantage of the proposed method when compared with GP_{SR} is that it does not need thresholds for selecting seeds to be used in the segmentation process. At the end of each relevance feedback interaction, the classifier itself defines the relevance level for all of the subimages. GP_{SR} is used as baseline in our experiments.

3 The *GOPF* Approach

The *GOPF* approach is a framework for recognition of regions of interest in remote sensing images combining *OPF* [11] classifier and *GP* [5] composite descriptor.

Optimum-path forest (OPF) is a classification method that represents each class of objects by one or more optimum-path trees rooted at given samples, called prototypes [11]. The training samples are nodes of a complete graph whose arcs are weighted by the distance between the feature vectors of their nodes.

The use of OPF for relevance feedback considers two classes: relevant subimages (chosen by the user) and irrelevant ones. The prototypes computed by the OPF classifier are used to rank database images according to the user's selection.

Algorithm 1 illustrates how $GOPF$ is used in the classification system. Let $\hat{I}$ be an RSI divided into a set of subimages (block regions). For every subimage $t \in \hat{I}$, we have a feature vector $\boldsymbol{v}(t) \in \Re^n$. That is, every subimage may be interpreted as a point in the feature space $\Re^n$. The distance $d(s,t)$ between two images s and t is the distance between their corresponding feature vectors. For an initial query point s, the proposed method returns the N closest subimages in $\hat{I}$ to s (query by similarity). Due to the semantic gap problem, the closest subimages to s may not be the most relevant for a given user. By marking the relevant subimages among the returned ones, the user creates two sets: a set $\mathcal{I} \subset \hat{I}$ of irrelevant subimages and a set $\mathcal{R} \subset \hat{I}$ of relevant subimages. The method then uses sets $\mathcal{R}$ and $\mathcal{I}$ to compute two optimum-path forests (OPF), one for each class. Each subimage $t \in \hat{I} \backslash \mathcal{I} \cup \mathcal{R}$ is then classified according to the root's label of the forest (relevant/irrelevant) that offers to t the optimum path in the graph. Only the N closest images labeled as relevant will be returned (in a set $\mathcal{C}$) to the user in the next iteration. Relevant prototypes ($\mathcal{A}$) and irrelevant ones ($\mathcal{B}$), computed in the previous step, are then used to sort the subimages in $\mathcal{C}$ for the next iteration.

After classifying each subimage in $\mathcal{I} \backslash \mathcal{I} \cup \mathcal{R}$, the method returns to the user a new set of N relevant subimages, which contains the lowest values of $\bar{d}(t, \mathcal{A}, \mathcal{B})$. This process is then repeated for a few iterations T and, finally, the system returns all relevant subimages obtained so far. The complete approach, which we call $GOPF$ (GP+OPF), is illustrated in Figure 1 (a).

The performance of the classifier is directly dependent on the good feature description of the objects involved in the classification. In order to combine spectral and texture features from various descriptors, the distance $d(s,t)$ between two images s and t used by OPF classifier is provided by a GP-based composite descriptor [5]. The GP framework requires a training set to find a good combination function. As the method is interactive we propose training GP with the prototypes ($\mathcal{A}$ and $\mathcal{B}$) provided by the OPF since it is a very informative subset of subimages.

The GP module starts with a population of combination functions created randomly. This population evolves generation by generation through genetic operations (e.g., crossover, mutation, reproduction). A fitness function is used to assign the fitness value for each individual based on the ranking of the training set. This value is used to select the best individuals. Next, genetic operators are applied to this population aiming to create more diverse and better performing individuals. The last step consists in determining the best individual to be applied to the test set. The commonest choice is the individual with the best performance in the training set (e.g., the first function of the last generation).

In this work we adopt the same notion of descriptor proposed in [5]. In this case, a GP individual is a function used to combine the distances provided by a set of single descriptors concerning the features extracted from two subimages.

Algorithm 1 The subimage recognition process in the *GOPF*.

Compute the mean distance $d(s,t)$ from the descriptors for every subimage $t \in \mathcal{Z}$.
Create an ordered list L of the N closest subimages t to s based on $d(s,t)$.
Set $\mathcal{I} \leftarrow \emptyset$ and $\mathcal{R} \leftarrow \emptyset$.
for each learning iteration $i = 1, 2, \ldots, T$ **do**
    Set $\mathcal{C} \leftarrow \emptyset$.
    The user marks the relevant subimages in L, which are inserted into $\mathcal{R}$ and the irrelevant ones are inserted into $\mathcal{I}$.
    if $|\mathcal{R}| < N$ **then**
        Compute OPF using sets $\mathcal{I}$ and $\mathcal{R}$, resulting also $\mathcal{A}$ and $\mathcal{B}$.
        for each subimage $t \in \mathcal{Z} \backslash \mathcal{I} \cup \mathcal{R}$ **do**
            if t is labeled as relevant by OPF **then**
                Insert t into the set $\mathcal{C}$ of images classified as relevant.
            end if
        end for
    else
        Show the final ordered list L with the N most relevant subimages in $\mathcal{R}$, as defined by the user selection.
    end if-else
    Create an ordered list L with the N most relevant subimages in $\mathcal{C}$, in increasing order of $\bar{d}(t, \mathcal{A}, \mathcal{B})$.
    Apply GP to find the distance combination function $f(d_i(s,t))$ by using $\mathcal{A}$ and $\mathcal{B}$ as training set.
    Recombine the subimages features by using the best GP function $f(d_i(s,t))$
end for
Return the final ordered list L with the N most relevant subimages in $\mathcal{R}$, completing it with the $N - |\mathcal{R}|$ relevant subimages in $\mathcal{C}$ in the increasing order of $\bar{d}(t, \mathcal{A}, \mathcal{B})$.

The GP individual configuration in the *GOPF* is the same as that used for GP_{SR} method [6]. Figure 1 (b) shows an example of GP individual as a function to combine features from two subimages. This individual corresponds to the function $f(d_1(s,t), d_2(s,t), d_3(s,t)) = \frac{d_1(s,t) * d_3(s,t)}{d_2(s,t)} - \sqrt{d_2(s,t) + d_3(s,t)}$.

4 Experiments

This section describes the experiments performed to validate our method. The experiment Setup is described as the following:

- **GP Parameters:** We used the values presented in Table 1.
- **Data:** Two RSIs were used to validade our method. Information about used RSIs is showed in Table 2. In this paper, we call Image 1 as PASTURE and image 2 as COFFEE.
- **Features:** We used the same set of descriptors described in [6]: BIC, Color Histograms, Color Moments, Gabor Wavelets, and Spline Wavelets.

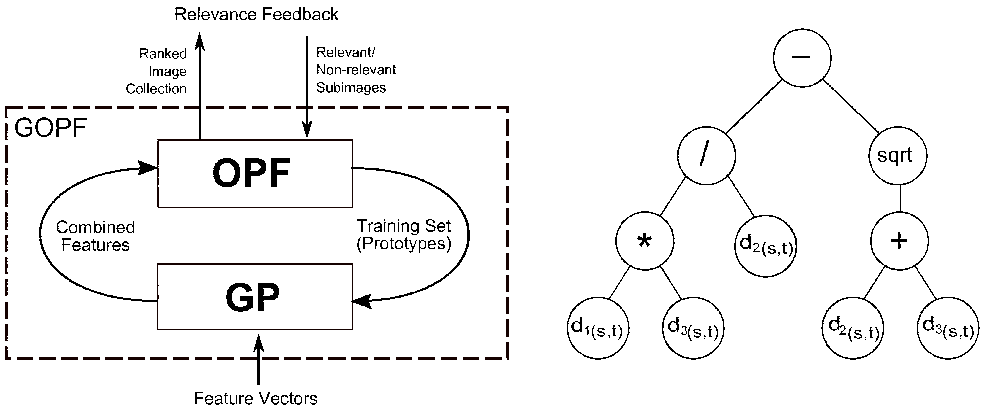


Fig. 1. The proposed interactive classifier (GOPF).

Table 1. GP Parameters in the *GOPF* framework.

population size	60
number of generations	10
initial population	*half-and-half*
initial tree depth	2 – 5
maximum tree depth	5
selection method	tournament (size 2)
crossover rate	0.8
mutation rate	0.2
functions set	$+, *, \sqrt{}, d^{const}$

- **Baseline:** We compare our method against Maximum Likelihood Classification (MaxVer) [12] and GP_{SR} [6]. PASTURE image was classified by *MaxVer* with probability threshold 0.8 and using 20,580 points of pasture sample. COFFEE image was classified with probability threshold 0.98 and using 43,630 points of coffee sample.
- **Effectiveness measure:** We use *kappa iterations* curves as effectiveness measure. Kappa [4] is an effective index to compare classified images, commonly used in RSI classification. Experiments in different areas show that kappa could have various interpretations and these guidelines could be different depending on the application. However, Landis and Koch [7] characterize Kappa values over 0.80 as "almost perfect agreement", 0.60 to 0.79 as "substantial agreement", 0.40 to 0.59 as "moderate agreement", and below 0.40 as "poor agreement". Kappa negative means that there is no agreement between classified and verification data.

Figure 2 (a) illustrates the curve kappa-iterations of the *GOPF* in comparison with GP_{SR} and the value found by the *MaxVer* for the PASTURE

Table 2. Remote Sensing Images used in the experiments.

	Image 1	**Image 2**
Region of interest	pasture	coffee
Terrain	plain	mountainous
Satellite	CBERS	SPOT
Spatial resolution	20 meters	2.5 meters
Bands composition	R-IR-G (342)	IR-NIR-R (342)
Acquisition date	08–20–2005	08–29–2005
Location	Laranja Azeda Basin, MS	Monte Santo County, MG
Dimensions (px)	1310×1842	2400×2400

image. Note that the value of kappa is always greater for the proposed method when compared to MaxVer, along iterations. Note also that the hybrid approach, $GOPF$, yields better results than GP_{SR}.

Figure 2 (b) illustrates the curve-kappa iterations of $GOPF$ in comparison with the kappa value found by GP_{SR} and $MaxVer$ for the COFFEE image. The kappa values for the proposed methods were better than the MaxVer score. Another remarkable result is concerned with the superiority of $GOPF$ when compared to GP_{SR}.

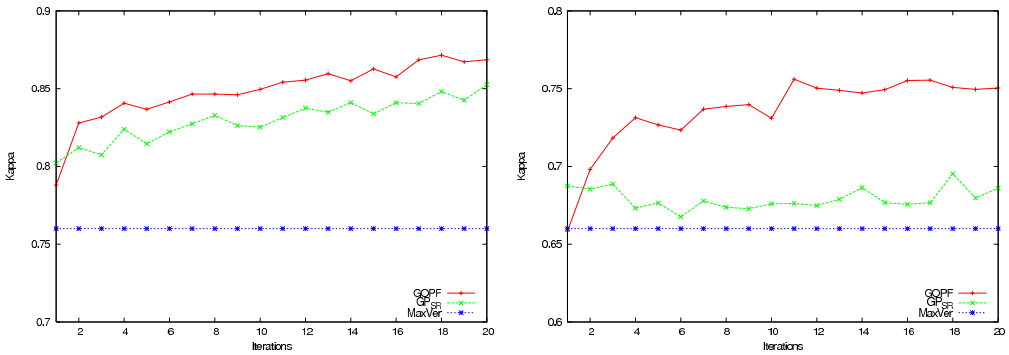


Fig. 2. Kappa X Iterations comparing the results of the proposed method, GP_{SR} and the MaxVer to the PASTURE image (a) and to the COFFEE image.

5 Conclusions

We have proposed a hybrid framework for recognition of regions of interest in remote sensing images which combines OPF [11] classifier and GP [5] composite descriptor. The system uses image descriptors to encode the spectral and texture regions of the RSIs and exploits user's relevance feedback. $GOPF$ has presented good results with respect to the identification of pasture and coffee

crops, overcoming the results obtained by GP_{SR} [6] and the MaxVer algorithm. As future works, we plan to evaluate more image descriptors; to allow user to define multiple regions as query pattern; and to compare the method with other baselines.

6 Acknowledgments

The authors are grateful to CAPES, FAPESP, and CNPq for financial support.

References

1. S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay. Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45:1506–1511, May 2007.
2. Bazi and Melgani. Toward an Optimal SVM Classification System for Hyperspectral Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 44:3374–3385, November 2006.
3. T. Blaschke. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):2 – 16, 2010.
4. Russel G. Congalton and Kass Green. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices.* Lewis Publishersr, Washington, DC, 1977.
5. R. da S. Torres, A.X. Falcão, M.A. Gonçalves, J.P. Papa, B. Zhang, W. Fan, and E.A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):217–312, Feb 2009.
6. J.A. dos Santos, C.D. Ferreira, R. da S.Torres, M.A. Gonçalves, and R.A.C. Lamparelli. A relevance feedback method based on genetic programming for classification of remote sensing images. *Information Sciences*, 181(13):2671 – 2684, 2011.
7. J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, March 1977.
8. N. Li, H. Huo, and T. Fang. A novel texture-preceded segmentation algorithm for high-resolution imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, (99):1 –11, 2010.
9. D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007.
10. J. Munoz-Mari, L. Bruzzone, and G. Camps-Valls. A Support Vector Domain Description Approach to Supervised Classification of Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 45:2683–2692, August 2007.
11. J.P. Papa, A.X. Falcão, and C.T.N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.
12. Robert Showengerdt. *Techniques for Image Processing and Classification in Remote Sensing.* Academic Press, New York, 1983.
13. Ming-Hseng Tseng, Sheng-Jhe Chen, Gwo-Haur Hwang, and Ming-Yu Shen. A genetic algorithm rule-based approach for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):202 – 212, 2008.
14. G.G. Wilkinson. Results and implications of a study of fifteen years of satellite image classification experiments. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):433–440, March 2005.

Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning

André Tavares da Silva[a], Jefersson Alex dos Santos[b], Alexandre Xavier Falcão[b], Ricardo da Silva Torres[b], Léo Pini Magalhães[a]

[a]*Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, University of Campinas (Unicamp), 400 Albert Einstein Avenue, 13083-970, Campinas, SP, Brazil*
[b]*Institute of Computing, University of Campinas (Unicamp), 1251 Albert Einstein Avenue, 13083-852, Campinas, SP, Brazil*

Abstract

In content-based image retrieval (CBIR) using feedback-based learning, the user marks the relevance of returned images and the system learns how to return more relevant images in a next iteration. In this learning process, image comparison may be based on distinct distance spaces due to multiple visual content representations. This work improves the retrieval process by incorporating multiple distance spaces in a method based on optimum-path forest (OPF) classification. For a given training set with relevant and irrelevant images, an optimization algorithm finds the best distance function to compare images as a combination of their distances according to different representations. Two optimization techniques are evaluated: a multi-scale parameter search (MSPS), never used before for CBIR, and a genetic programming (GP) algorithm. The combined distance function is used to project an OPF classifier and to rank images classified as relevant for the next iteration. The ranking process takes into account relevant and irrelevant representatives, previously found by the OPF classifier. Experiments show the advantages in effectiveness of the proposed approach with both optimization techniques over the same approach with single distance space and over another state-of-the-art method based on multiple distance spaces.

Email addresses: atavares@dca.fee.unicamp.br (André Tavares da Silva), jsantos@ic.unicamp.br (Jefersson Alex dos Santos), afalcao@ic.unicamp.br (Alexandre Xavier Falcão), rtorres@ic.unicamp.br (Ricardo da Silva Torres), leopini@fee.unicamp.br (Léo Pini Magalhães)

Keywords: Content-Based Image Retrieval, Optimum-Path Forest Classifiers, Composite Descriptor, Genetic Programming, Multi-Scale Parameter Search, Image Pattern Analysis.

1. Introduction

Large image collections have increased the demand for efficient and effective information retrieval methods based on the visual content of the images. The simplest visual content representation is a *feature vector*, which encodes color, texture, and/or shape measures of an image. The similarity between images can be measured by the *distance* among their feature vectors. For a given query image, a content-based image retrieval (CBIR) system aims to return the most relevant images with respect to the query. However, a *semantic gap* usually occurs between this result and the user's expectation, due to the absence of relevant information in the feature vectors. In order to reduce the semantic gap problem, feedback-based learning approaches have been investigated [1]. In these methods, the user indicates which images are relevant (irrelevant) in a small set of returned images and the CBIR system learns how to return more relevant images in a next iteration. This search process can be repeated until the user is satisfied.

The visual content representations may be based on local [2, 3, 4] and/or global [5, 6, 7, 8, 9] image properties. Some feedback-based methods learn how to select the best features, assuming a single distance space [10, 11]. However, multiple image representations may require distinct distance functions [7, 8, 9, 12, 13, 14]. Therefore, some approaches provide a fixed combination of the distance values between images that use multiple representations [15] and more elaborated methods learn the combined distance function that maximizes the effectiveness of the CBIR system [16, 17, 18, 19, 20, 21]. Other methods exploit the set of relevant and irrelevant images, as indicated by the user, to change the query point [22], to create multiple query points [23, 24, 25], or to design a pattern classifier [26, 27, 28, 29, 30, 31]. In the last case, database images classified as relevant are ranked and presented to the user for a next iteration. The classifier-based methods can also be divided into those that return the most relevant images at each iteration [32, 33, 31] and those that return the most informative images (i.e., images that are difficult to be classified) during a few iterations [16, 26, 27, 28, 29, 30], postponing the most relevant ones to the last iteration. While the strategy

of the former approaches is *greedy*, aiming the most relevant images at each iteration, the strategy of the latter methods is *planned*, in the sense that the user has to decide in which iteration the system will return the most relevant images. Planned approaches are widely known as *active learning* methods.

Methods that combine distance spaces are more complete and tend to be more effective in most applications [19, 20, 21]. In addition, other works have demonstrated that the classification of the relevant candidates before ranking is a good strategy to increase the effectiveness of the CBIR system [29, 30, 31]. We have proposed a greedy approach based on optimum-path forest (OPF) classification [31] and demonstrated its gain in effectiveness with respect to the first active learning method based on support vector machine (SVM) [26]. The choice of the OPF classifier is also justified by its considerable gain in efficiency with respect to SVM and other classification models, such as artificial neural networks and k-nearest neighbors [34].

In this paper we considerably improve our previous work [31] by extending it to handle multiple distance spaces. Two optimization techniques are evaluated in this task: a multi-scale parameter search (MSPS) [35], never used for CBIR, and a genetic programming (GP) algorithm [18]. The choice of GP is also motivated by the successful recent works [21, 36]. Santos et al. [36], for example, showed that the GP-based method improves the retrieval effectiveness, outperforming methods based on genetic algorithm [21].

For a given training set with relevant and irrelevant images, the best combined distance function to compare images based on multiple representations is found by optimization, using MSPS or GP. The training set is interpreted as a complete graph weighted on the arcs by the combined distance between nodes. The closest images between the relevant and irrelevant classes are chosen as representative images (*prototypes*). The prototypes compete among themselves and each prototype conquers its most strongly connected images, according to a *path-cost function*, partitioning the training set into an optimum-path forest (classifier). For any database image, the OPF classifier efficiently assigns the label of its most strongly connected prototype in an incremental way. Images classified as relevant are ranked in their increasing order of a normalized distance (always using the combined distance function) with respect to the relevant and irrelevant prototypes. Thus, the most relevant images are presented to the user for a next iteration of relevance feedback.

This paper is organized as follows. Section 2 reviews the original CBIR approach based on relevance feedback and optimum-path forest classifica-

tion [31]. The method is described for single distance spaces and then the optimization techniques that integrate multiple distance spaces are presented in Section 3. Exhaustive experiments involving several datasets, two state-of-the-art approaches [31, 21], visual content representations with their distance functions, and the two optimization techniques are presented in Section 4. Finally, Section 5 states conclusion.

2. Relevance feedback with optimum-path forest

Let $\mathcal{Z}$ be an image database, such that each image t is represented by a feature vector $\vec{v}(t)$, for sake of simplicity. We define a *simple descriptor* D as a pair (v, d), being v the feature extraction function and $d(s, t)$ the distance function between two image representations (e.g. $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$) [37]. The extension to multiple descriptors is presented in Section 3.

In content-based image retrieval (CBIR), we aim to return a list $\mathcal{X}$ with the N most relevant images in $\mathcal{Z}$ with respect to a given query image q. The simplest approach is to return the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q, t)$. However, due to the limitation of the descriptor (v, d) in representing the user's expectation (*semantic gap*), the list $\mathcal{X}$ contains relevant and irrelevant images according to the user's opinion. To reduce the semantic gap problem, the system asks for the user's feedback about the relevance of the returned images during a few iterations. The user indicates which images are relevant (or irrelevant) in $\mathcal{X}$, forming a *labeled training set* $\mathcal{T}$ which gains new elements at each iteration of relevance feedback by $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$.

This section describes the feedback-based learning process, named OPF_{RF}, using optimum-path forest (OPF) classification [31]. In OPF_{RF}, set $\mathcal{T}$ is used to project a new OPF classifier [34] at each iteration. This process consists of first estimating representative samples (*prototypes*) in $\mathcal{T}$ for each image class, forming the sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of relevant and irrelevant prototypes, respectively. The training process considers a complete graph, whose nodes are all elements in $\mathcal{T}$ and arcs (s, t) between images s and t are weighted by $d(s, t)$. Every path in the graph has a cost and minimum-cost paths are computed from $\mathcal{S}_R \cup \mathcal{S}_I$ to each node $t \in \mathcal{T}$, such that the classifier is an optimum-path forest rooted in $\mathcal{S}_R \cup \mathcal{S}_I$ (Section 2.1). In this forest, the nodes $t \in \mathcal{T} \backslash \mathcal{S}_R \cup \mathcal{S}_I$ are conquered and labeled (in the same class as relevant/irrelevant) by the prototype in $\mathcal{S}_R \cup \mathcal{S}_I$ which offers the optimum path with terminus t. Afterward, this classifier evaluates the images in $\mathcal{Z} \backslash \mathcal{T}$, by computing the cost of the optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$ to each node $t \in \mathcal{Z} \backslash \mathcal{T}$

in an incremental way, and inserts t in a reduced set $\mathcal{Y} \subset \mathcal{Z} \backslash \mathcal{T}$ of *relevant candidates* whenever the optimum path is rooted in $\mathcal{S}_R$.

The system then returns a new list $\mathcal{X} \subset \mathcal{Y}$ with the N closest images in the increasing order of a normalized mean distance $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ between t and the two sets of prototypes (Equation 1) [31].

$$\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I) = \frac{\bar{d}(t, \mathcal{S}_R)}{\bar{d}(t, \mathcal{S}_R) + \bar{d}(t, \mathcal{S}_I)} \quad (1)$$

$$\bar{d}(t, \mathcal{S}_R) = \frac{1}{|\mathcal{S}_R|} \sum_{\forall s \in \mathcal{S}_R} d(s, t), \quad (2)$$

$$\bar{d}(t, \mathcal{S}_I) = \frac{1}{|\mathcal{S}_I|} \sum_{\forall s \in \mathcal{S}_I} d(s, t). \quad (3)$$

Again, the user may indicate relevant and irrelevant images in $\mathcal{X}$ and a new training set is created by $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ to redesign an improved classifier. This process is repeated until the user is satisfied.

Next Sections detail the optimum-path forest training and classification processes and the feedback-based learning algorithm, respectively.

2.1. Training and Classification by Optimum-Path Forest

Given a complete graph, whose nodes are all images in $\mathcal{T}$, a path π_t in the graph with terminus t is a sequence $\langle t_1, t_2, \ldots, t_n = t \rangle$ of distinct nodes. The *strength of connectedness* between the origin $R(\pi_t) = t_1$ and the terminus $t_n = t$ is inversely proportional to maximum arc weight $d(t_i, t_{i+1})$ (i.e., weakest link) along it, as defined by a *path-cost function*

$$c(\pi_t) = \max_{i=1,2,\ldots,n-1} \{d(t_i, t_{i+1})\}. \quad (4)$$

A path π_t is optimum when $c(\pi_t) \leq c(\tau_t)$ for any other path τ_t with the same terminus t. For every node $t \in \mathcal{T}$, we wish to compute a minimum-cost path with terminus t and origin $R(t) \in \mathcal{S}_R \cup \mathcal{S}_I$. The idea is to obtain an optimum partition of $\mathcal{T}$ such that the relevant and irrelevant prototypes in $\mathcal{S}_R$ and $\mathcal{S}_I$ will compete with each other and every training node $t \in \mathcal{T}$ will be assigned to the class of its most strongly connected prototype. This partition is computed as an optimum-path forest P — an acyclic graph which stores all optimum paths in a predecessor map P, i.e., a function which assigns to

every node $t \in \mathcal{T} \backslash \mathcal{S}_R \cup \mathcal{S}_I$ its predecessor node $P(t)$ in the optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$, or a mark $P(t) = nil$ when $t \in \mathcal{S}_R \cup \mathcal{S}_I$.

Algorithm 1 presents the computation of an optimum-path forest P for path-cost function c [38, 34] (Equation 4) constrained to start in $\mathcal{S}_R \cup \mathcal{S}_I$. It outputs for each node $t \in \mathcal{T}$, the minimum cost $C(t)$ of the optimum path with terminus t, its root node $R(t) \in \mathcal{S}_R \cup \mathcal{S}_I$, and the predecessor $P(t)$ in that optimum path. A list $\mathcal{T}'$ of the training nodes in the increasing order of cost $C(t)$ is also output to speedup classification.

Algorithm 1: OPF Algorithm

Input: Training set $\mathcal{T}$, the sets of prototypes $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$, and a descriptor (v, d).
Output: Optimum-path forest P (predecessor map), path-cost map C, root map R, and the ordered list $\mathcal{T}'$ of training nodes.
Auxiliary: Priority queue Q and cost variable *cst*.

```
1  for each s ∈ T\S_R ∪ S_I do
2      Set C(s) ← +∞
3  end
4  for each s ∈ S_R ∪ S_I do
5      Set C(s) ← 0, P(s) ← nil,
6      R(s) ← s, and insert s into Q.
7  end
8  while Q is not empty do
9      Remove from Q a node s with minimum C(s) and insert s in T'.
10     for each t ∈ T such that C(t) > C(s) do
11         Compute cst ← max{C(s), d(s,t)}.
12         if cst < C(t) then
13             if C(t) ≠ +∞ then
14                 remove t from Q.
15             end
16             P(t) ← s, R(t) ← R(s) and C(t) ← cst.
17             Insert t in Q.
18         end
19     end
20 end
```

Given that the optimum paths for function c (Equation 4) tend to choose the closest nodes to form each link along the paths, the sets of prototypes $\mathcal{S}_R$ and $\mathcal{S}_I$ must be chosen from the closest samples between the relevant and irrelevant classes. These prototypes tend to avoid misclassification by an optimum path coming from the opposite class. Let $\lambda(t)$ be the class of image $t \in \mathcal{T}$, which may be relevant or irrelevant. The prototypes are obtained by computing a *minimum spanning tree* (MST) in the complete graph with nodes in $\mathcal{T}$ [39]. For each arc (s,t) in the MST, if $\lambda(s) \neq \lambda(t)$ then s and t are marked as prototypes. If $\lambda(s)$ is relevant and $\lambda(t)$ is irrelevant, then s is inserted in $\mathcal{S}_R$ and t is inserted in $\mathcal{S}_I$. Similarly, it is done in the other way around.

The algorithm follows the dynamic programming scheme [38]. Lines 1-7 initialize the cost, predecessor, and root maps, forcing the optimum paths to start in $\mathcal{S}_R \cup \mathcal{S}_I$, and insert the roots in Q. The main loop computes an optimum path from $\mathcal{S}_R \cup \mathcal{S}_I$ to every node $s \in \mathcal{T}$ in a non-decreasing order of cost (Lines 8–20). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q, preserving its order in $\mathcal{T}'$ (Line 9). The rest of the lines evaluate if the path that reaches an adjacent node $t \neq s$ through s is cheaper than the current path with terminus t and update Q, $C(t)$, $R(t)$, and $P(t)$ accordingly.

In the classification phase, for every sample $t \in \mathcal{Z} \backslash \mathcal{T}$, the optimum path with terminus t can be easily identified by finding which training node $s^* \in \mathcal{T}$ provides the minimum value in

$$C(t) = \min_{\forall s \in \mathcal{T}} \{\max\{C(s), d(s,t)\}\}. \tag{5}$$

Node s^* is the predecessor $P(t)$ in the optimum path with terminus t and the image t is classified as $\lambda(R(s^*))$.

The role of $\mathcal{T}'$, sorted by the cost $C(s)$ computed by Algorithm 1, is to speed up the evaluation of Equation 5. The search for the minimum value can be halted when the cost for a node p whose position in $\mathcal{T}'$ succeeds the position of s is greater then the cost computed previously ($C(p) > \max\{C(s), d(s,t)\}$) [40]. Thus, it is not necessary to compare each sample $t \in \mathcal{Z} \backslash \mathcal{T}$ with all elements of the set $\mathcal{T}$.

2.2. Feedback-based learning algorithm

Algorithm 2 presents the feedback-based learning using OPF classification [31]. It returns a list $\mathcal{R} \subset \mathcal{Z}$ of images in the decreasing order of relevance.

Algorithm 2: OPF_{RF} Algorithm

Input: A query image q, a descriptor (v, d), the number N of returned images per iteration, and an image database $\mathcal{Z}$.

Output: A list $\mathcal{R} \subset \mathcal{Z}$ of retrieved images in the decreasing order of relevance.

Auxiliary: Set $\mathcal{T}$ of training images, maps $(R, P, C, \mathcal{T}')$ of the OPF classifier, sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of prototypes, set $\mathcal{Y} \subset \mathcal{Z} \backslash \mathcal{T}$ of images classified as relevant, and ordered list $\mathcal{X} \subset \mathcal{Y}$ of N returned images per iteration.

1 Set $\mathcal{R} \leftarrow \emptyset$ and $\mathcal{T} \leftarrow \emptyset$.
2 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q, t)$.
3 **while** *the user is not satisfied* **do**
4 The user marks relevant (irrelevant) images, forming a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ where each image has a relevance label $\lambda(t)$.
5 Insert in $\mathcal{R}$ the relevant images of $\mathcal{X}$.
6 Compute sets $\mathcal{S}_R$ and $\mathcal{S}_I$ of prototypes in $\mathcal{T}$ (Section 2.1) and execute $(R, P, C, \mathcal{T}') \leftarrow \text{OPF}(\mathcal{T}, \mathcal{S}_R, \mathcal{S}_I, v, d)$ (Algorithm 1).
7 Classify images in $\mathcal{Z} \backslash \mathcal{T}$ using $(R, P, C, \mathcal{T}')$ and $\lambda(t)$ for $t \in \mathcal{S}_R \cup \mathcal{S}_I$ (Equation 5), and create set $\mathcal{Y}$ with the relevant candidates.
8 Compute a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ in the increasing order of $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$. (Equation 1)
9 **end**
10 Return $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{X}$.

In Line 1, the output list $\mathcal{R}$ and the training set $\mathcal{T}$ are initialized to empty sets. For a given query image q and image database $\mathcal{Z}$, the algorithm first returns a list $\mathcal{X}$ with the N closest images $t \in \mathcal{Z}$ in the increasing order of $d(q, t)$ (Line 2). The learning process is executed in the main loop (Lines 3-9). In Line 4, the user marks the relevant images in $\mathcal{X}$, and the remaining images are considered as irrelevant, or vice-verse. This essentially creates a training set $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{X}$ where each image $t \in \mathcal{T}$ has a relevance label $\lambda(t)$. The relevant images of $\mathcal{X}$ are inserted in the output list $\mathcal{R}$ (Line 5). The OPF training is computed in Line 6 (Algorithm 1). It results sets $\mathcal{S}_R \subset \mathcal{T}$ and $\mathcal{S}_I \subset \mathcal{T}$ of relevant and irrelevant prototypes; the OPF attributes, predecessor $P(t)$, cost $C(t)$, and root $R(t)$ for each $t \in \mathcal{T}$; and set $\mathcal{T}'$ of training images in the increasing order of optimum cost $C(t)$ for faster classification. In Line

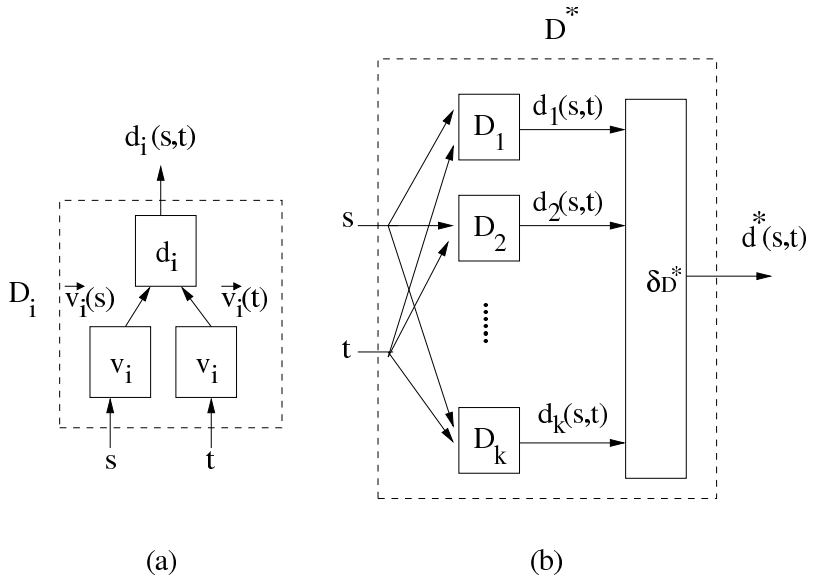


Figure 1: (a) Simple and (b) Composite descriptor $D^* = (\mathcal{D}, \delta D^*)$ [37].

7, OPF is used to classify images in $\mathcal{Z} \backslash \mathcal{T}$ using Equation 5, forming a set $\mathcal{Y}$ with the images classified as relevant. A new list $\mathcal{X}$ with the N closest images $t \in \mathcal{Y}$ is created in the increasing order of a *normalized mean distance* $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$ between t and the two sets of prototypes.

The main loop repeats until the user is satisfied and the retrieved images in $\mathcal{R}$ are all relevant images together with the last set $\mathcal{X}$ in the increasing order of $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.

3. Extension to multiple descriptors

Now consider a set $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ of simple descriptors such that a distance $d_i(s,t)$, $i = 1, 2, \ldots, n$, between any given pair of images $s, t \in \mathcal{Z}$ can be computed as a function of their respective representations $\vec{v}_i(s)$ and $\vec{v}_i(t)$.

We wish to find the best combination of simple descriptors in order to retrieve relevant images from $\mathcal{Z}$ with respect to a given query image and user. This combination is found through a *composite descriptor* [37] that integrates the respective distance spaces. That is, a composite descriptor D^* is a pair $(\mathcal{D}, \delta D^*)$, where δD^* is a function that best combines the distance values computed by each descriptor into a final distance value $d^*(s,t)$ (Figure 1).

Observe that the distance values from different descriptors may diverge significantly in scale. Thus, it is important that all values be normalized between 0 and 1. A Gaussian normalization [10] can be employed for that.

At each iteration of the feedback-based learning process, a training set $\mathcal{T}$ with relevant and irrelevant images is used to choose the best combination function using some optimization algorithm. For a given $\mathcal{T}$ and set $\mathcal{D}$ of descriptors, the optimization process essentially evaluates candidates δD according to some criterion function. The criterion function should measure the ability of the system in ranking the most relevant images first. We measure this property by averaging the FFP4 [41] values (Equation 6) with respect to each relevant image in $\mathcal{T}$, rather than using only the query image. This approach has proven to be more effective than the simplest method based on the query image only [18]. Let $\mathcal{T}_R$ be the subset of relevant images in $\mathcal{T}$ and $\mathcal{T}_u$ be a set of training images $t \in \mathcal{T}$ in their increasing order of distance $\delta D(t,u)$ with respect to a given relevant image $u \in \mathcal{T}_R$. Thus, our criterion function $F(\mathcal{T},\mathcal{D},\delta D)$ is defined by

$$F(\mathcal{T},\mathcal{D},\delta D) \;=\; \frac{1}{\mid \mathcal{T}_R \mid} \sum_{\forall u \in \mathcal{T}_R} \sum_{k=1}^{|\mathcal{T}_u|} 7\lambda_k 0.982^k, \tag{6}$$

where $\lambda_k \in \{0,1\}$ indicates the user's opinion about the relevance of each image at every position k in $\mathcal{T}_u$, as either relevant ($\lambda_k = 1$) or irrelevant ($\lambda_k = 0$).

In Algorithm 2, the complete graph used to compute the OPF classifier has its arcs (s,t) weighted by the best combined distance $d^*(s,t)$ that results from the optimization process. The best combination function δD^* is also used to compute the normalized distance $\bar{d}(t,\mathcal{S}_R,\mathcal{S}_I)$ in that algorithm.

In the following sections we present two optimization techniques to compute composite descriptors: the multi-scale parameter search (MSPS) and a genetic programming algorithm (GP).

3.1. Composite descriptors using Multi-Scale Parameter Search

In the multi-scale parameter search (MSPS), the combination function δD is represented by a same equation and the parameters of this function must be optimized [35]. In this paper, this equation is defined by

$$\delta D(s,t) \;=\; \sum_{i=1}^{n} d_i^{\theta_i}(s,t) \tag{7}$$

where $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ is its parameter vector, being $0 \leq \theta_i \leq 2$. The parameters may give distinct non-linear emphases to their corresponding descriptors. Given that the values of $F(\mathcal{T}, \mathcal{D}, \delta D)$ during optimization will depend only on the choice of θ, we will denote $F(\mathcal{T}, \mathcal{D}, \delta D)$ by $F(\mathcal{T}, \mathcal{D}, \theta)$ for clarity. From any given state $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, the idea is to find the best displacement vector $\mathbf{\Delta}^*$, by exploiting multiple scales of the parameter space, and update the parameter vector into a next value $\theta \leftarrow \theta + \mathbf{\Delta}^*$, repeating this process until a maximum $F(\mathcal{T}, \mathcal{D}, \theta^*)$. In order to avoid local maxima, the method perturbs θ on each parameter axis $i = 1, 2, \ldots, n$ and with m displacement scales $j = 1, 2, \ldots, m$ along each axis. At each iteration, it evaluates $F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta})$ for the displacement vectors $\mathbf{\Delta}$ that result from all perturbations on each axis, separately, all scales, and the resulting vectors from each scale. This method works as follows.

Let $0 \leq \Delta_{i,j} \leq 1$ be a positive displacement along the parameter axis i for a scale j. The method takes into account the following displacements:

- the best perturbation along each parameter axis i, as $\mathbf{\Delta}^*_{i,j} = (0, \ldots, \Delta^*_{i,j}, \ldots, 0)$ for $\Delta^*_{i,j} \in \{\Delta_{i,j}, 0, -\Delta_{i,j}\}$, such that

$$F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta}^*_{i,j}) = \max \left\{ \begin{array}{l} F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta}_{i,j}), \\ F(\mathcal{T}, \mathcal{D}, \theta), \\ F(\mathcal{T}, \mathcal{D}, \theta - \mathbf{\Delta}_{i,j}) \end{array} \right\} \quad (8)$$

- and the resulting vectors $\mathbf{\Delta s}_j = \sum_{i=1}^{n} \mathbf{\Delta}^*_{i,j}$, $j = 1, 2, \ldots, m$.

Hence, the choice of $\mathbf{\Delta}^*$ can be expressed by:

$$F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta}^*) = \max \left\{ \begin{array}{ll} F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta}^*_{i,j}) & \text{for } i\text{=1,2,...,}n \text{ and} \\ & j\text{=1,2,...,}m. \\ F(\mathcal{T}, \mathcal{D}, \theta + \mathbf{\Delta s}_j) & \text{for } j\text{=1,2,...,}m. \end{array} \right\} \quad (9)$$

Algorithm 3 illustrates the computation of the best parameter vector θ^* in Equation 7. In this work, we fixed the displacements $\Delta_{i,j}$ as 0.001, 0.01, 0.12, 0.45, and 1.0 along $m = 5$ scales for every parameter $i = 1, 2, \ldots, n$. These displacements could have also be created by some increasing function. In Line 1, vector θ is initialized with the same weight to all descriptors. Lines 4–16 search for the θ^* that maximizes Equation 6. If there is a $\mathbf{\Delta}$ (Line 8), tested for positive and negative displacements, that increases the current

criterion value, it is stored in $\boldsymbol{\Delta}^*$ (Lines 10–11). If a better displacement vector is found, $\boldsymbol{\Delta}^*$ is updated in Line 12. Lines 15–16 test if $\boldsymbol{\Delta}\mathbf{s}$ generates a better combination function. The algorithm stops when no combination better than θ is found, returning in Line 19 the last parameter vector θ^*, which is then used in Equation 7 to define the final composite descriptor for feedback-based learning using the OPF classifier.

Algorithm 3: MSPS Algorithm

Input: Training set $\mathcal{T}$, set $\mathcal{D}$ of simple descriptors, and displacements $\Delta_{i,j}$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
Output: The best parameter vector θ^*.
Auxiliary: Displacement vectors $\boldsymbol{\Delta}$, $\boldsymbol{\Delta}^*$ and $\boldsymbol{\Delta}\mathbf{s}$, parameter vectors θ^* and θ, and variables i, j, V_0, V^-, V^+, and V^*.

1 $\theta \leftarrow (1, \ldots, 1)$;
2 $V^* \leftarrow F(\mathcal{T}, \mathcal{D}, \theta)$ and $\theta^* \leftarrow \theta$;
3 **repeat**
4 $V_0 \leftarrow V^*$ and $\theta \leftarrow \theta^*$;
5 **for** *j=1 to m* **do**
6 $\boldsymbol{\Delta}\mathbf{s} \leftarrow (0, \ldots, 0)$;
7 **for** *i=1 to n* **do**
8 $\boldsymbol{\Delta} \leftarrow (0, \ldots, \Delta_{i,j}, \ldots, 0)$, $V \leftarrow V_0$, and $\boldsymbol{\Delta}^* \leftarrow (0, \ldots, 0)$;
9 $V^+ \leftarrow F(\mathcal{T}, \mathcal{D}, \theta + \boldsymbol{\Delta})$ and $V^- \leftarrow F(\mathcal{T}, \mathcal{D}, \theta - \boldsymbol{\Delta})$;
10 **If** $V^+ > V$ **then** $V \leftarrow V^+$ and $\boldsymbol{\Delta}^* \leftarrow \boldsymbol{\Delta}$;
11 **If** $V^- > V$ **then** $V \leftarrow V^-$ and $\boldsymbol{\Delta}^* \leftarrow -\boldsymbol{\Delta}$;
12 **If** $V > V^*$ **then** $\theta^* \leftarrow \theta + \boldsymbol{\Delta}^*$ and $V^* \leftarrow V$;
13 $\boldsymbol{\Delta}\mathbf{s} \leftarrow \boldsymbol{\Delta}\mathbf{s} + \boldsymbol{\Delta}^*$;
14 **end**
15 $V \leftarrow F(\mathcal{T}, \mathcal{D}, \theta + \boldsymbol{\Delta}\mathbf{s})$;
16 **If** $V > V^*$ **then** $V^* \leftarrow V$ and $\theta^* \leftarrow \theta + \boldsymbol{\Delta}\mathbf{s}$;
17 **end**
18 **until** $V^* > V_0$
19 Return θ^*.

3.2. Composite descriptors using Genetic Programming

In the genetic programming (GP) approach [42], the combination function δD is a tree of mathematical operations applied to the distance values that result from the simple descriptors (Figure 2). Leaf nodes are represented

by these distance values, while the mathematical operatiors are defined in the internal nodes. Figure 2 exemplifies an individual with three distinct descriptors using the set $\{+, -, /, sqrt\}$ of operators as internal nodes. The best combination function $\delta D^*(s,t)$ is $\frac{d_1(s,t)+d_3(s,t)}{d_2(s,t)} - \sqrt{d_2(s,t)+d_3(s,t)}$ in this example.

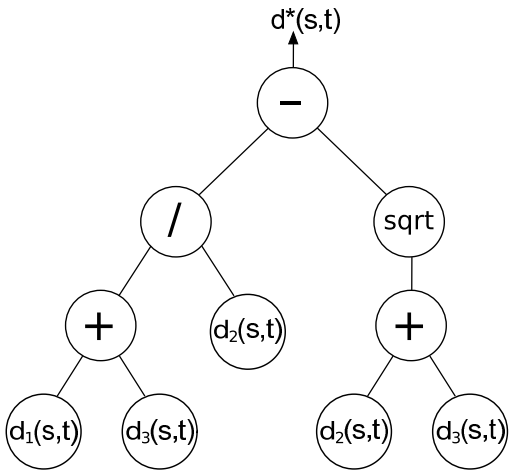


Figure 2: A tree representing the best combination function of a composite descriptor.

The GP approach [18] searches for the best combination function by evolving a population $\mathcal{P}$ of n_p candidate individuals per iteration during n_g generations. At each iteration, all individuals δD_i, $i = 1, 2, \ldots, n_p$, are evaluated by function $F(\mathcal{T}, \mathcal{D}, \delta D_i)$ (Equation 6). The best individuals are identified and modified by applying genetic transformations, such as reproduction, mutation, and crossover. The reproduction operator copies them to the next generation. Mutation randomly selects a point in the GP tree of an individual and replaces the subtree of that point with a new randomly generated subtree. The crossover operator swaps a subtree of one parent with a subtree of another [42].

Algorithm 4 describes the search for the best individual δD^* during n_g generations, starting from an initial population $\mathcal{P}$ with n_p individuals. The population starts with individuals created randomly (Line 1). This population evolves generation by generation through genetic operations (Lines 2–9). The criterion function $F(\mathcal{T}, \mathcal{D}, \delta D_i)$ (Equation 6) is used to assign the fitness value for each individual (Line 5). This value is used to select the best individuals (Line 7). Next, genetic operators are applied to this population in

order to create more diverse and better performing individuals (Line 8). The last step consists of determining the best individual (Line 10). The commonest choice is the individual with the best performance in the last generation of $\mathcal{P}$.

Algorithm 4: GP Algorithm

Input: A training set $\mathcal{T}$, a set $\mathcal{D}$ of descriptors, the size n_p of the population, the number n_g of generations, and percentage of reproduction, mutation and crossover.
Output: The best individual δD^* (a tree structure).
Auxiliary: Set $\mathcal{P}$ (population) of n_p individuals δD_i, a set $\mathcal{A}$ of pairs $(\delta D_i, F(\mathcal{T}, \mathcal{D}, \delta D_i))$, and variables i and g.

1 $\mathcal{P} \leftarrow$ Initial random population of n_p individuals;
2 $\mathcal{A} \leftarrow \emptyset$;
3 **for** *each generation* $g = 1, 2, \ldots, n_g$ **do**
4 **for** *each individual* $\delta D_i \in \mathcal{P}$, $i = 1, 2, \ldots, n_p$ **do**
5 Insert $\mathcal{A} \leftarrow \mathcal{A} \cup (\delta D_i, F(\mathcal{T}, \mathcal{D}, \delta D_i))$;
6 **end**
7 Sort $\mathcal{A}$ in the decreasing order of $F(\mathcal{T}, \mathcal{D}, \delta D_i)$;
8 Create a new population $\mathcal{P}$ of size n_p by reproduction, crossover and mutation among the best individuals in $\mathcal{A}$;
9 **end**
10 Return the best individual δD^* in $\mathcal{A}$ with the highest value of $F(\mathcal{T}, \mathcal{D}, \delta D^*)$.

4. Experiments and Results

We call OPF_{MSPS} and OPF_{GP} the feedback-based learning process using OPF classification and the optimization techniques MSPS and GP respectively.

Table 1 shows the values used for GP parameters [42] used in the OPF_{GP} feedbach-based learning method. Table 2 presents the population size and the number of generations of GP in the OPF_{GP} for each dataset.

Table 1: Parameter values for GP in the OPF_{GP} method.

initial population	*half-and-half*
initial tree depth	2 – 5
maximum tree depth	5
selection method	tournament (size 2)
crossover prob.	0.8
mutation prob.	0.9
reproduction prob.	0.05
functions set	$+, *, \sqrt{}$

Table 2: Population size and number of generations of GP in the OPF_{GP} method for each dataset.

	n_p	n_g
Coil100	40	8
Corel3906	100	10
Eth80	100	10
Mpeg7	100	10
msrcorid	50	10
Pascal	60	10

We compare the effectiveness of OPF_{MSPS} and OPF_{GP} against OPF_{RF}, as proposed in [31] and described in Section 2 for simple descriptors (the best one for each dataset), and another approach, named GP^+ [21]. GP^+ is a feedback-based learning method which uses genetic programming to compute the best composite descriptor given a training set composed only by relevant (positive) images. GP^+ ranks the database images by a voting scheme among the best individuals of the last population. In this voting scheme, selected individuals vote for N candidate images. The most voted images are showed to the user. GP^+ outperforms several other feedback-based learning techniques [10, 22, 26, 43, 44] as well as the ranking process based on the best individual d^* obtained by Algorithm 4 with no OPF classification. In our

experiments, the values for the GP parameters are the same as those used for the Corel Collection in [21].

For each image database, we simulate the user behavior by using each image as initial query point and marking the relevant points (images from the same class of the query) among 30 returned images per iteration.

We use the precision-recall ($P \times R$) curve for 3, 5 and 8 iterations to measure effectiveness. We use the $P \times R$ curve considering the results obtained at the last RF iteration. Higher the curve, better the method. The first comparison evaluates the importance of having a composite descriptor while the second comparison evaluates the importance of having a pattern classifier.

We also present the percentage of relevant images retrieved to the user given a number of relevance feedback iterations of OPF_{MSPS}, OPF_{GP}, and GP^{+}. This curve allows us to evaluate how well the number of retrieved relevant images grows over iterations. Both curves show the average result considering all database images computed using the leave-one-out cross-validation. Since the curves use the entire image database $\mathcal{Z}$, line 10 of Algorithm 2 is replaced by: *Return* $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{Y}$ *in their increasing order of* $\bar{d}(t, \mathcal{S}_R, \mathcal{S}_I)$.

The experiments used six heterogeneous image databases, representing different challenges for CBIR.

- Coil-100 [45]

  It is an image database of 100 objects. Pictures of each object were taken in 72 different poses (total of 7,200 images).

- Corel [46]

  This database is a collection with 200,000 images from the Corel Gallery Magic–Stock Photo Library 2. We use a subset of 3,906 samples, preclassified into 85 classes. These classes have different number of images varying from 7 to 98 images each.

- ETH-80 [47].

  This database is available in the project COGVIS. The project includes images of objects from 8 basic-level categories performing a total of 2,384 images, distributed uniformly among the classes.

- MPEG7 (MPEG7 CE Shape-1 Part B) [48]

  It is a database of 1,400 binary images of 70 shape categories, being 20 objects per category.

- MSRCORID [49]

  It contains a set of 4,320 images grouped into 20 categories – about 36 to 652 per category. Most categories have about 200 images.

- PASCAL [50]

  This database consists of images from Flickr [1]. We use a subset of 3,448 grouped into 23 classes with different number of images, varying from 72 to 446 subimages each.

We use several image descriptors to test the methods. The results presented in this paper make use of the following descriptors: ACC [5], BIC [7], Color Bitmap [51], Fourier [52], JAC [9], LAS [6], LBP [53], MSF (Multi-Scale Fractal) [12], SASI [8], SID [13] and TSDIZ [14]. ACC, BIC, Color Bitmap and JAC are *color* descriptors, while LAS, LBP, SASI and SID are *texture* descriptors. Fourier, Multi-Scale Fractal and TSDIZ are used for databases with *shape* information.

For each image database, we chose the best simple descriptors for combination, as shown in Table 3 (except for the case of Coil-100, we chose the worst ones, because the problem was too easy with the best one – BIC [7]). For Corel, MSRCORID and PASCAL databases we selected three color descriptors and two texture descriptors. In ETH-80 we used shape, color, and texture descriptors. In MPEG7 we used all shape descriptors above.

Figures 3 to 11 show the mean precision-recall curves of each method in all image databases for 3, 5 and 8 iterations of relevance feedback . These images show the evolution in the performance of each method. One may observe that both OPF_{MSPS} and OPF_{GP} outperformed GP^{+} in all tested databases. OPF_{MSPS} outperformed OPF_{GP} in Coil-100 and Pascal image databases while OPF_{GP} was more effective than OPF_{MSPS} in ETH-80 and MSRCORID databases. In Corel database, OPF_{MSPS} had better result up to the fifth iteration and OPF_{GP} beated OPF_{MSPS} after eight iterations. In MPEG7 shape database, OPF_{MSPS} had a good result for three iterations

[1] www.flickr.com

Table 3: Descriptors to be combined in each database.

Database	Descriptors
Coil-100	SID, LBP and Color Bitmap
Corel	ACC, BIC, JAC, LAS and SASI
ETH-80	ACC, BIC, LAS, MSF and TSDIZ
MPEG7	Fourier, MSF and TSDIZ
MSRCORID	ACC, BIC, JAC, LAS and SASI
PASCAL	ACC, BIC, JAC, LAS and SASI

while OPF_{GP} outperforms OPF_{MSPS} from five iterations. In addition to that, OPF_{RF} using the best descriptor (TSDIZ for the MPEG7 database and BIC for the others) was more effective than GP^{+} in most results, proving that the classification step is very important in the feedback-based learning process. Figures 12 to 14 present the mean curves of relevant returned images per iteration ($Rel \times It$) for all databases from the first to the eighth iteration, confirming the previous results.

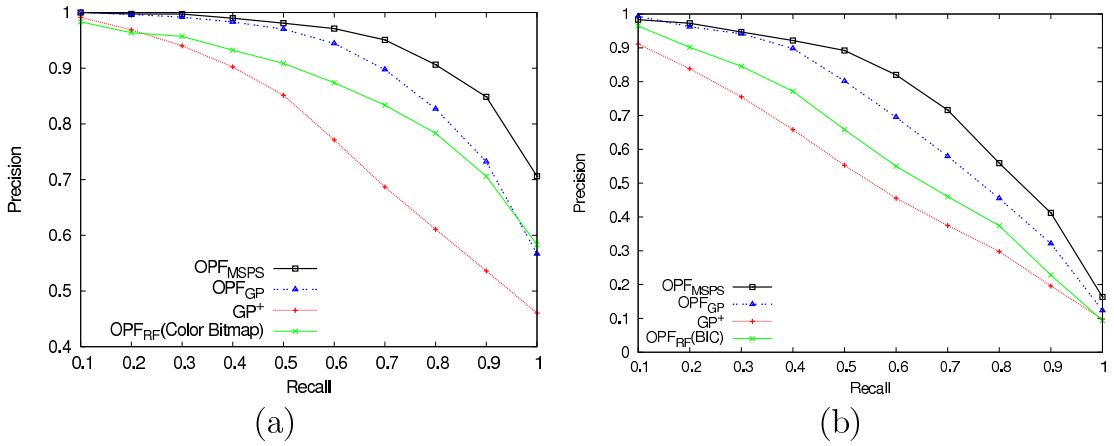


Figure 3: Mean $P \times R$ curves in (a) Coil-100 and (b) Corel databases after 3rd iteration.

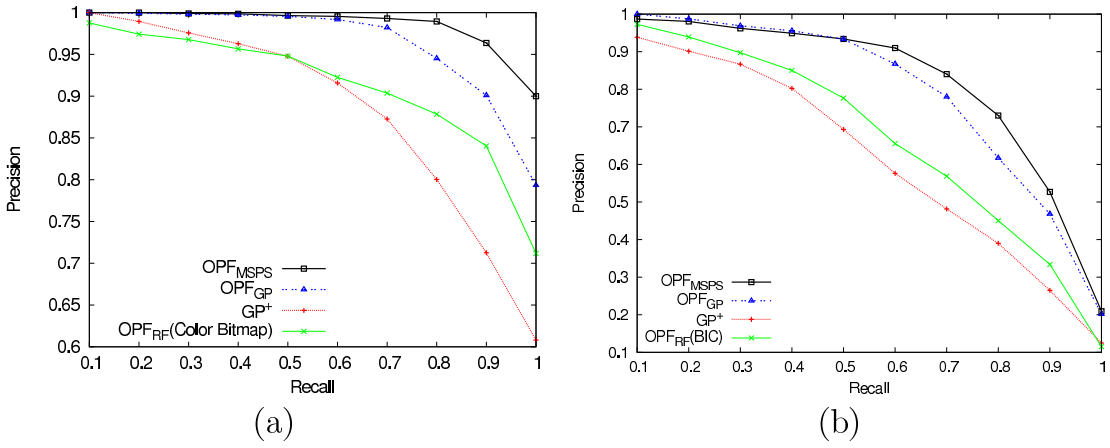


Figure 4: Mean $P \times R$ curves in (a) Coil-100 and (b) Corel databases 5th iteration.

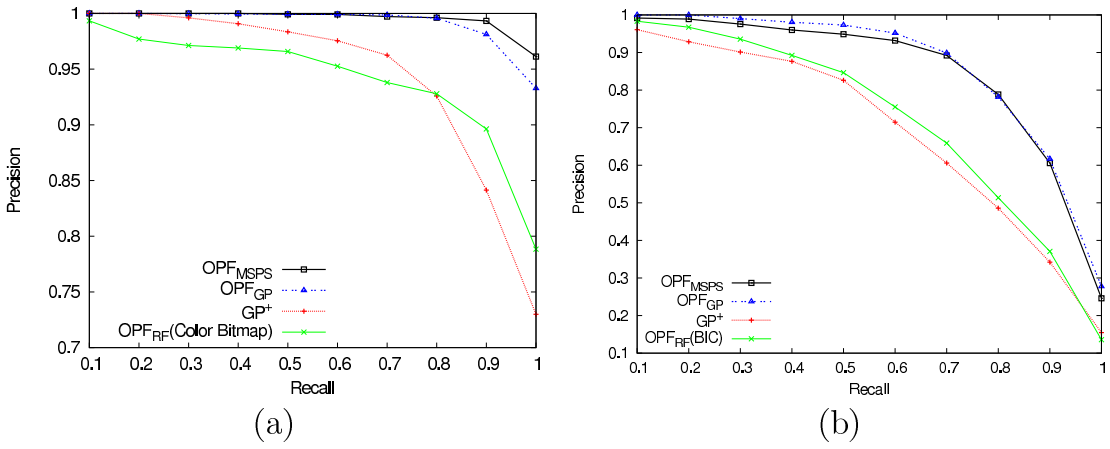


Figure 5: Mean $P \times R$ curves in (a) Coil-100 and (b) Corel databases after 8th iteration.

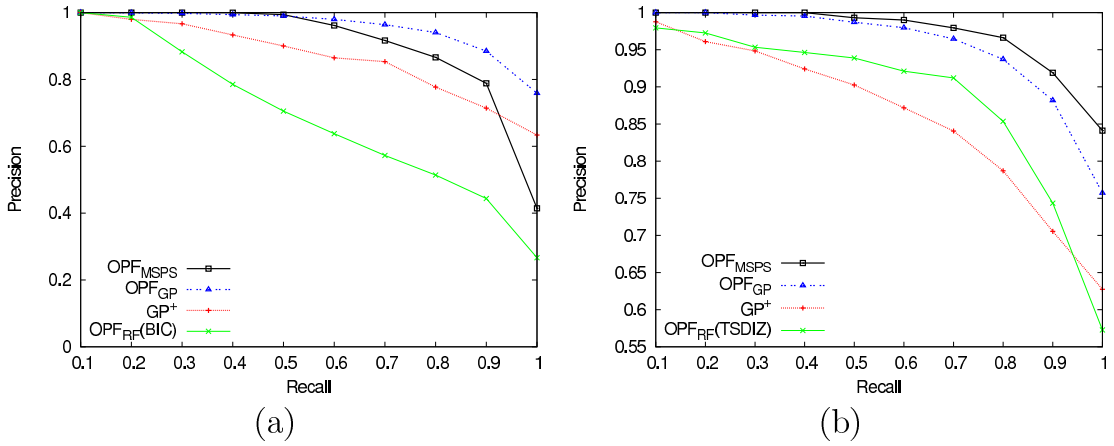


Figure 6: Mean $P \times R$ curves in (a) ETH-80 and (b) MPEG7 databases after 3rd iteration.

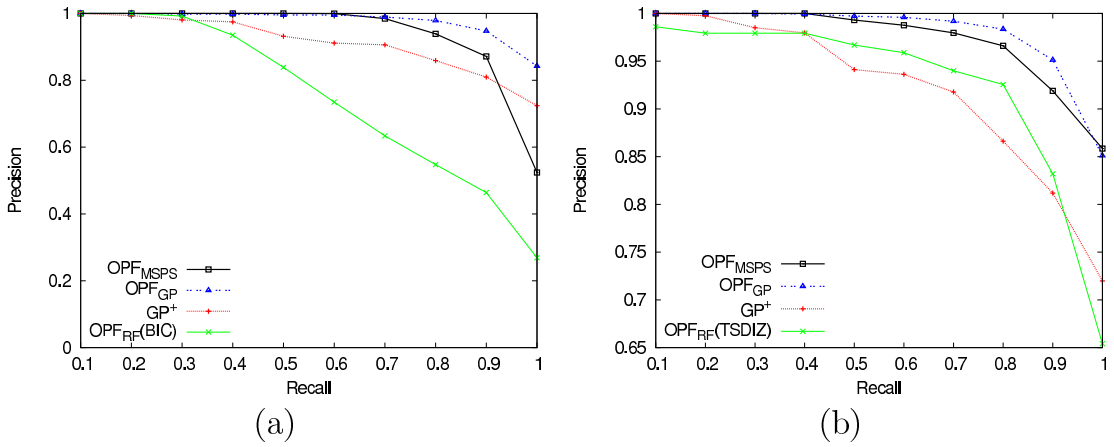


Figure 7: Mean $P \times R$ curves in (a) ETH-80 and (b) MPEG7 databases after 5th iteration.

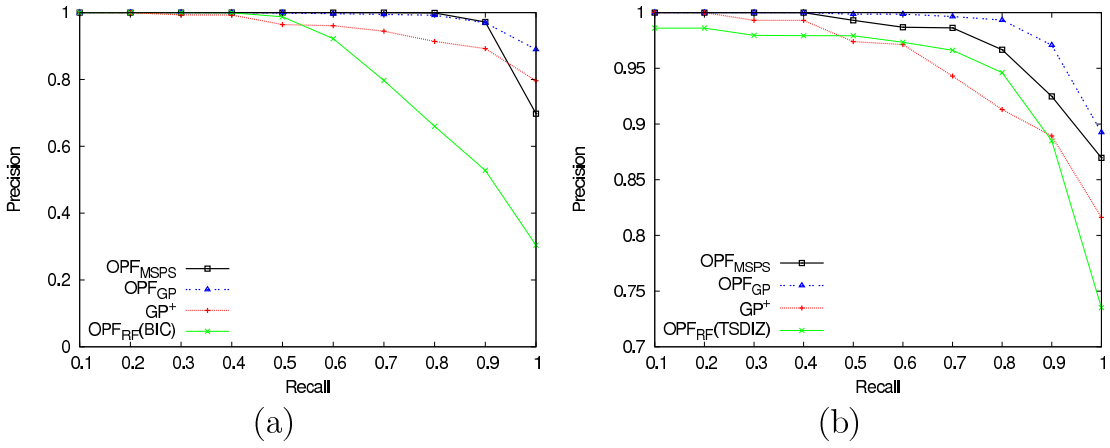


Figure 8: Mean $P \times R$ curves in (a) ETH-80 and (b) MPEG7 databases after 8th iteration.

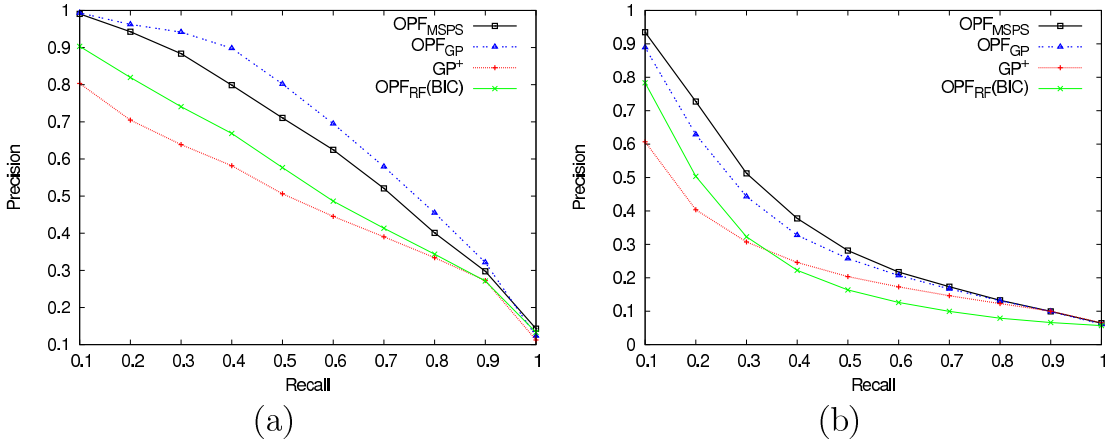


Figure 9: Mean $P \times R$ curves in (a) MSRCORID and (b) Pascal databases after 3rd iteration.

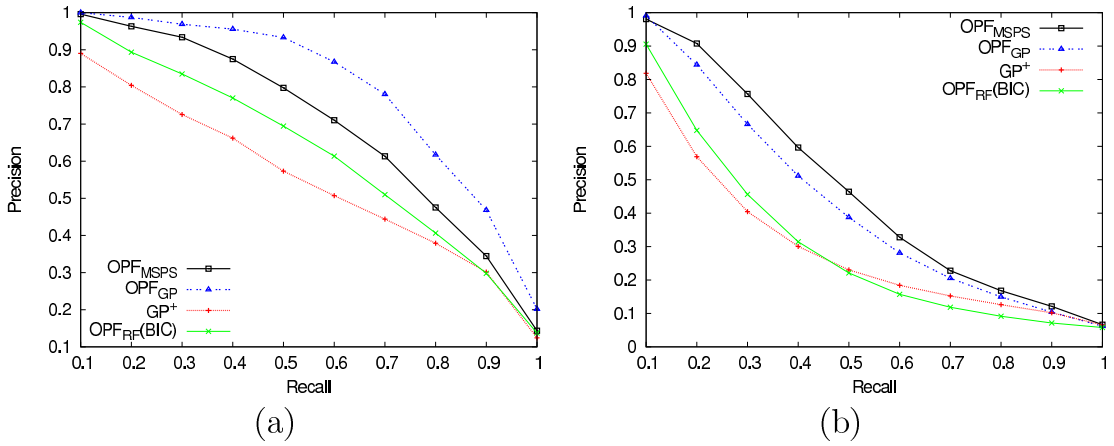


Figure 10: Mean $P \times R$ curves in (a) MSRCORID and (b) Pascal databases after 5th iteration.

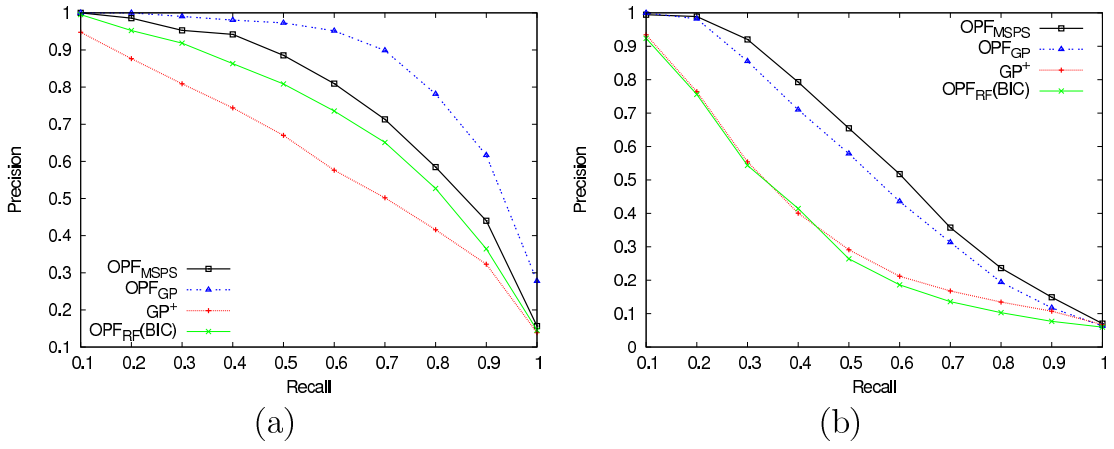


Figure 11: Mean $P \times R$ curves in (a) MSRCORID and (b) Pascal databases after 8th iteration.

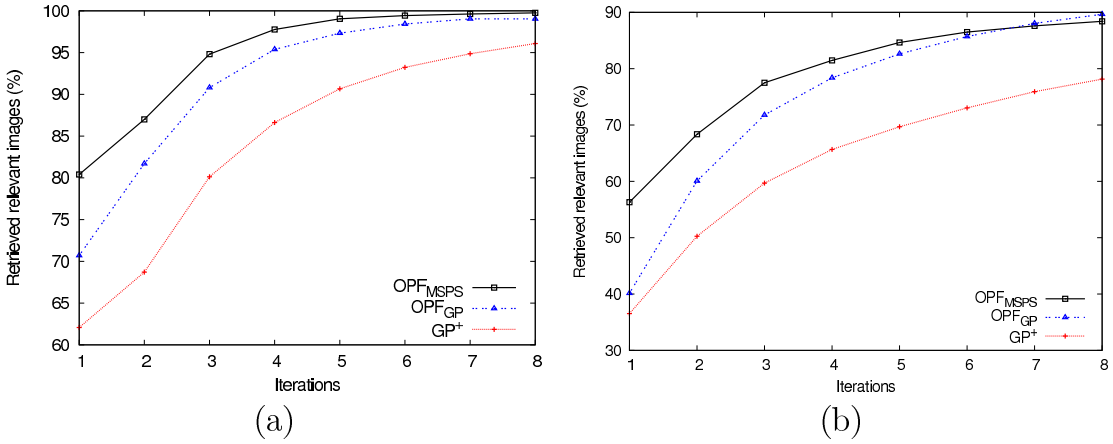


Figure 12: $Rel \times It$ to (a) Coil-100 and (b) Corel databases from 1st to 8th iteration.

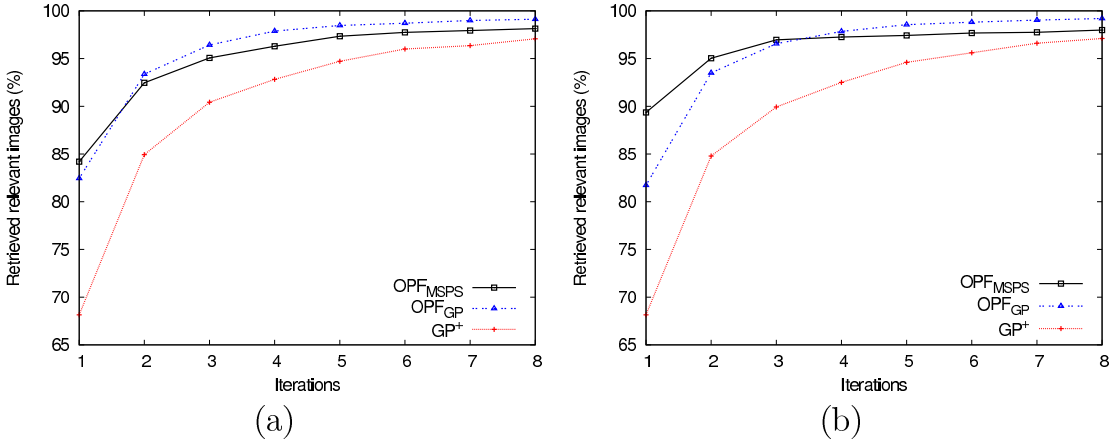


Figure 13: $Rel \times It$ to (a) ETH-80 and (b) MPEG7 databases from 1st to 8th iteration.

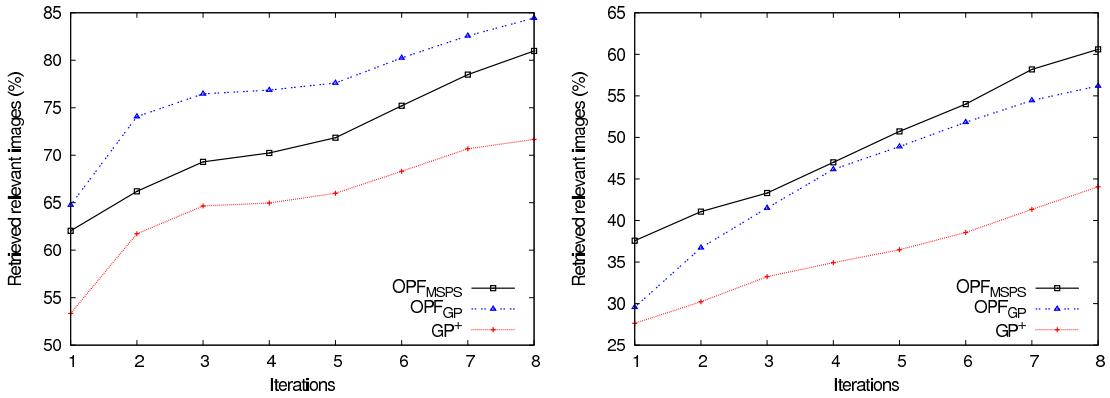


Figure 14: $Rel \times It$ to (a) MSRCORID and (b) Pascal databases from first to eighth iteration.

5. Conclusion

We extended a recent feedback-based learning approach for CBIR using OPF classification (OPF_{RF}) to handle composite descriptors. The new methods, OPF_{MSPS} and OPF_{GP}, use optimization techniques, such as multi-scale parameter seach (MSPS) and genetic programming (GP), to find the best combination function for a given set of simple descriptors at each iteration of relevance feedback.

Experiments with several datasets and descriptors have demonstrated that both OPF_{MSPS} and OPF_{GP} are very effective methods, with better performance than other state-of-the-art approaches, GP^+ and OPF_{RF}. These experiments also show the importance of using multiple descriptors and patter classifiers for CBIR. The best choice between OPF_{MSPS} and OPF_{GP} will depend on the dataset (application). However, it is important to note that MSPS assumes a fixed equation for the combination function and searches for its best parameters, while GP is more flexible in the choice of the best combination function. On the other hand, MSPS is faster than GP, and this favors to the choice of OPF_{MSPS}. The execution time of MSPS varies according to the number of descriptors to be combined. The number of scales has some influence, but tests have indicated that lower number of scales usually imply a higher number of iterations to achieve the best combination function.

A drawback in GP is its sensitivity to the initial parametrization (values of n_p and n_g for instance). This requires a previous study of the impact of

each parameter for each dataset. Its efficiency depends on the number of generations, population size, training set size, and the size of the individuals [41].

The results also indicate that OPF_{GP} and OPF_{MSPS} require a few iterations of relevance feedback to retrieve the most relevant images.

Our future work involves the study of indexing schemes to provide scalability in large image databases with fast image access. We also intend to consider irrelevant images in the training set used by GP in the OPF_{GP} method.

Acknowledgements

The first author thanks CNPq for financial support (140968/2007-5). The third author thanks CNPq (481556/2009-5, 302617/2007-8) and FAPESP (2007/52015-0, 2008/57428-4).

References

[1] X. S. Zhou, T. S. Huang, Relevance feedback in image retrieval: A comprehensive review, Multimedia Systems 8 (6) (2003) 536–544.

[2] D. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, 1999, pp. 1150–1157.

[3] I. Laptev, B. Caputo, C. Schldt, T. Lindeberg, Local velocity-adapted motion events for spatio-temporal recognition, Computer Vision and Image Understanding 108 (3) (2007) 207–229, special Issue on Spatiotemporal Coherence for Visual Motion Analysis.

[4] H. Bay, T. Tuytelaars, L. V. Gool, Surf: Speeded up robust features, in: Computer Vision and Image Understanding (CVIU), Vol. 110, 2008, pp. 346–359.

[5] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on (1997) 762–768.

[6] B. Tao, B. W. Dickinson, Texture recognition and image retrieval using gradient indexing, Journal of Visual Communication and Image Representation 11 (3) (2000) 327–342.

[7] R. O. Stehling, M. A. Nascimento, A. X. Falcão, A compact and efficient image retrieval approach based on border/interior pixel classification, in: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, ACM, New York, NY, USA, 2002, pp. 102–109.

[8] A. ÇarkacIoglu, F. Yarman-Vural, Sasi: a generic texture descriptor for image retrieval, Pattern Recognition 36 (11) (2003) 2615 – 2633.

[9] A. Williams, P. Yoon, Content-based image retrieval using joint correlograms, Multimedia Tools and Applications 34 (2) (2007) 239–248.

[10] Y. Rui, T. S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: A power tool for interactive content-based image retrieval, in: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8 (5), 1998, pp. 644–655.

[11] A. Vadivel, A. Majumdar, S. Sural, Characteristics of weighted feature vector in content-based image retrieval applications, in: Intelligent Sensing and Information Processing, no. 18 in 1, 2004, pp. 127–132.

[12] R. S. Torres, A. X. Falcão, L. F. Costa, A graph-based approach for multiscale shape analysis, Pattern Recognition 37 (6) (2004) 1163–1174.

[13] J. A. Montoya-Zegarra, N. J. Leite, R. da S.Torres, Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval, in: SIBGRAPI '07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society, Washington, DC, USA, 2007, pp. 121–128.

[14] F. A. Andaló, P. A. V. Miranda, R. S. Torres, A. X. F. ao, Shape feature extraction and description based on tensor scale, Pattern Recognition 43 (1) (2010) 26–36.

[15] R. Dorairaj, K. Namuduri, Compact combination of mpeg-7 color and texture descriptors for image retrieval, Conference Record of the Thirty-Eighth Asilomar Conference on Signals 1 (38) (2004) 387–391.

[16] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, P. N. Yianilos, The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments, IEEE Transactions on Image Processing 1 (9) (2000) 20–37.

[17] M. L. Kherfi, D. Brahmi, D. Ziou, Combining visual features with semantics for a more effective image retrieval, in: ICPR '04: Proceedings of the Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2004, pp. 961–964.

[18] R. Torres, A. Falcão, M. Gonçalves, J. Papa, B. Zhang, W. Fan, E. Fox, A genetic programming framework for content-based image retrieval, Pattern Recognition 42 (2) (2009) 283–292.

[19] M. Arevalillo-Herráez, F. J. Ferri, J. Domingo, A naive relevance feedback model for content-based image retrieval using multiple similarity measures, Pattern Recognition 43 (3) (2010) 619–629.

[20] M. Broilo, F. De Natale, A stochastic approach to image retrieval using relevance feedback and particle swarm optimization, Multimedia, IEEE Transactions on 12 (4) (2010) 267–277.

[21] C. Ferreira, J. Santos, R. da S. Torres, M. Gonçalves, R. Rezende, W. Fan, Relevance feedback based on genetic programming for image retrieval, Pattern Recognition Letters 32 (1) (2011) 27 – 37.

[22] Y. Rui, T. Huang, Optimizing learning in image retrieval, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2000, pp. 236–243.

[23] K. Porkaew, K. Chakrabarti, S. Mehrotra, Query refinement for multimedia similarity retrieval in mars, in: Proceedings of ACM Multimedia, 1999, pp. 235–238.

[24] Z. hua Zhou, K. jia Chen, H. bin Dai, Enhancing relevance feedback in image retrieval using unlabeled data, ACM Transactions on Information Systems 24 (2006) 219–244.

[25] D. Liu, K. A. Hua, K. Vu, N. Yu, Fast query point movement techniques for large CBIR systems, IEEE Transactions on Knowledge and Data Engineering 21 (5) (2009) 729–743.

[26] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia, ACM, New York, NY, USA, 2001, pp. 107–118.

[27] T. Qin, X.-D. Zhang, T.-Y. Liu, D.-S. Wang, W.-Y. Ma, H.-J. Zhang, An active feedback framework for image retrieval, Pattern Recognition Letters 29 (2008) 637–646.

[28] X. Wang, L. Yang, Application of svm relevance feedback algorithms in image retrieval, in: Proceedings of the 2008 International Symposium on Information Science and Engieering, IEEE Computer Society, Washington, DC, USA, 2008, pp. 210–213.

[29] P.-H. G. S. Philipp-Foliguet, J. Gony, Frebir: An image retrieval system based on fuzzy region matching, Computer Vision and Image Understanding 113 (6) (2009) 693–707.

[30] S. C. H. Hoi, R. Jin, J. Zhu, M. R. Lyu, Semisupervised svm batch mode active learning with applications to image retrieval, ACM Transactions on Information Systems 27 (3) (2009) 1–29.

[31] A. T. Silva, A. X. Falcão, L. P. Magalhães, A new CBIR approach based on relevance feedback and optimumpath forest classification, Journal of WSCG 18 (1-3) (2010) 73–80.

[32] I. King, Z. Jin, Integrated probability function and its application to content-based image retrieval by relevance feedback, Pattern Recognition 36 (9) (2003) 2177–2186.

[33] G. Giacinto, F. Roli, Bayesian relevance feedback for content-based image retrieval, Pattern Recognition 37 (7) (2004) 1499–1508.

[34] J. P. Papa, A. X. Falcão, C. T. N. Suzuki, Supervised pattern classification based on optimum-path forest, International Journal of Imaging Systems and Technology 19 (2) (2009) 120–131.

[35] G. Ruppert, F. Favretto, A. Falcão, C. Yassuda, F. Bergo, Fast and accurate image registration using the multiscale parametric space and grayscale watershed transform, in: Systems, Signals and Image Processing, IEEE Computer Society, Rio de Janeiro, 2010, pp. 17–19.

[36] J. A. Santos, C. D. Ferreira, R. S. Torres, A genetic programming approach for relevance feedback in region-based image retrieval systems, in: SIBGRAPI '08: Proceedings of the 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society, 2008, pp. 155–162.

[37] R. S. Torres, A. X. Falcão, Content-based image retrieval: Theory and applications, Revista de Informática Teórica e Aplicada 13 (2) (2006) 161–185.

[38] A. Falcão, J. Stolfi, R. Lotufo, The image foresting transform: Theory, algorithms, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 19–29.

[39] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT, 1990.

[40] J. P. Papa, F. A. M. Cappabianco, A. X. Falcão, Optimizing optimum-path forest classification for huge datasets, in: Proceedings of The 20th International Conference on Pattern Recognition, 2010.

[41] W. Fan, E. A. Fox, P. Pathak, H. Wu, The effects of fitness functions on genetic programming-based ranking discovery for web search, Journal of the American Society for Information Science and Technology 55 (2004) 2004.

[42] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), 1st Edition, The MIT Press, 1992.

[43] N. Doulamis, A. Doulamis, Evaluation of relevance feedback schemes in content-based in retrieval systems, Signal Processing: Image Communication 21 (4) (2006) 334–357.

[44] R. Min, H. D. Cheng, Effective image retrieval using dominant color descriptor and fuzzy support vector machine, Pattern Recognition 42 (1) (2009) 147–157.

[45] S. A. Nene, S. K. Nayar, H. Murase, Columbia university image library (coil-100), http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php.

[46] J. Z. Wang, J. Li, G. Wiederhold, Simplicity: Semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 947–963.

[47] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 409–415.

[48] T. Sikora, The mpeg-7 visual standard for content description-an overview, Circuits and Systems for Video Technology, IEEE Transactions on 11 (6) (2001) 696–702.

[49] M. R. Cambridge, Microsoft research cambridge. object recognition image database 1.0, http://research.microsoft.com/vision/cambridge/recognition/.

[50] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge 2010 (voc2010), http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/index.html.

[51] T.-C. Lu, C.-C. Chang, Color image retrieval technique based on color features and image bitmap, Information Processing and Management 43 (2) (2007) 461–472, special issue on AIRS2005: Information Retrieval Research in Asia.

[52] D. Zhang, G. Lu, A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures, Journal of Visual Communication and Image Representation 1 (14) (2003) 41–60.

[53] V. Takala, T. Ahonen, M. Pietikäinen, Block-based methods for image retrieval using local binary patterns, in: Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA), 2005, pp. 882–891.